The Dissertation Committee for Andrew Jesse Mills
certifies that this is the approved version of the following dissertation:

# Lower Bounds and Correctness Results for Locally Decodable Codes

Committee:

Anna Gál, Supervisor

Adam Klivans

C. Greg Plaxton

Sriram Vishwanath

David Zuckerman

# Lower Bounds and Correctness Results for Locally Decodable Codes

**by**

**Andrew Jesse Mills, B.S.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2011

# Acknowledgments

I would like to thank my family, friends, and advisor. With regards to technical matters, I am especially appreciative of discussions with Mahdi Cheraghchi, William Gasarch, Rahul Jain, Jaikumar Radhakrishnan, and David Woodruff.

# Lower Bounds and Correctness Results for Locally Decodable Codes

Publication No. _____

Andrew Jesse Mills, Ph.D.
The University of Texas at Austin, 2011

Supervisor: Anna Gál

We study fundamental properties of Locally Decodable Codes (LDCs). LDCs are motivated by the intuition that traditional codes do not have a good tradeoff between resistance to arbitrary error and probe complexity. For example, if you apply a traditional code on a database, the resulting codeword can be resistant to error even if a constant fraction of it was corrupted; however, to accomplish this, the decoding procedure would typically have to analyze the entire codeword. For large data sizes, this is considered computationally expensive. This may be necessary even if you are only trying to recover a single bit of the database! This motivates the concept of LDCs, which encode data in such a way that up to a constant fraction of the result could be corrupted; while the decoding procedures only need to read a sublinear, ideally constant, number of codeword bits to retrieve any bit of the input with high probability.

Our most exciting contribution is an exponential lower bound on the length of three query LDCs (binary or linear) with high correctness. This is

the first strong length lower bound for any kind of LDC allowing more than two queries. For LDCs allowing three or more queries, the previous best lower bound, given by Woodruff, is below $\Omega(n^2)$. Currently, the best upper bound is sub-exponential, but still very large. If polynomial length constructions exist, LDCs might be useful in practice. If polynomial length constructions do not exist, LDCs are much less likely to find adoption – the resources required to implement them for large database sizes would be prohibitive. We prove that in order to achieve just slightly higher correctness than the current best constructions, three query LDCs (binary or linear) require exponential size.

We also prove several impossibility results for LDCs. It has been observed that for an LDC that withstands up to a $\delta$ fraction of error, the probability of correctness cannot be arbitrarily close to 1. However, we are the first to estimate the largest correctness probability obtainable for a given $\delta$. We prove close to tight bounds for arbitrary numbers of queries.

# Table of Contents

**Acknowledgments** **iv**

**Abstract** **v**

**Chapter 1. Introduction** **1**
1.1 Applications . . . . . 1
1.2 The Central Mystery . . . . . 7

**Chapter 2. General Facts about LDCs** **8**
2.1 Smooth Codes . . . . . 13
2.2 Overview of Previous Results . . . . . 17
2.3 Previous LDC Impossibility Results . . . . . 18
2.4 Previous LDC Lower Bounds Results . . . . . 19
2.5 Previous LDC Constructions . . . . . 22
2.6 Previous PIR Results . . . . . 24

**Chapter 3. Main Results Completed** **28**

**Chapter 4. Basic Impossibility Results** **35**

**Chapter 5. Technical Tools** **45**
5.1 Properties of Query Sets for Linear Codes . . . . . 45
5.2 Probabilistic Adversary . . . . . 49
5.3 Linear Decoders . . . . . 59

**Chapter 6. Length Lower Bounds for Three Query LDCs** **67**
6.1 Three Query, Binary, Linear LDCs . . . . . 67
6.2 Three Query, Binary, Possibly Non-Linear LDCs . . . . . 77
6.3 Three Query, Linear LDCs over Any Field . . . . . 86

**Chapter 7. Length Lower Bounds for Four Query, Linear, Binary LDCs 106**

**Chapter 8. Arbitrary Number of Queries for Special Classes of Decoders 133**

**Chapter 9. General Correctness Bounds 138**

9.1 Linear Algebra Property . . . . . 138
9.2 q Query, Binary, Linear LDCs . . . . . 141
9.3 Probabilistic Method for the Non-Linear Case . . . . . 151
9.4 q Query, Binary, Possibly Non-Linear LDCs . . . . . 155
9.5 q Query, Possibly Non-Linear LDCs Over Any Field . . . . . 166

**Chapter 10. Precise Bounds on Correctness for Two and Three Query LDCs 175**

10.1 Two Query, Binary, Linear LDCs . . . . . 175
10.2 Three Query, Binary, Linear LDCs . . . . . 180
10.3 Two Query, Binary, Possibly Non-Linear LDCs . . . . . 199

**Chapter 11. Locally Decodable Erasure Codes 205**

11.1 Definitions and Properties . . . . . 205
11.2 Correctness Bounds for Probabilistic LDECs . . . . . 210

**Chapter 12. Hadamard Codes Can Have Very Small Error 212**

**Bibliography 216**

**Vita 224**

# Chapter 1

# Introduction

Locally decodable codes (LDCs) are very interesting combinatorial structures, both in their own right and in the other structures that can be derived from them. They have many applications, including some in theoretical work and some in practical, "real world" work. Informally (from [42]), a $(q, \delta, \epsilon)$-LDC is a mapping of strings $\mathbf{C}$: $\Sigma^n \rightarrow \Gamma^m$ such that, given the string $\mathbf{C}(x_1 x_2 ... x_n)$ corrupted in at most $\delta m$ positions, each $x_i$ ($i \in [n]$) can be recovered with probability at least $\frac{1}{|\Sigma|} + \epsilon$ by examining at most $q$ positions of that string. This thesis, because of the motivations given shortly, focuses on proving fundamental properties of LDCs with $q \geq 2$.

## 1.1 Applications

LDCs were first introduced by Katz and Trevisan [32]. [32] considered the application of encoding a huge amount of data in such a way that up to a constant fraction of the result could be arbitrarily corrupted; however, the decoding procedure would only have to read a sublinear, ideally constant, number of codeword bits to retrieve any bit of the input with high probability. The intuition behind this is that traditional codes do not have a good tradeoff

between resistance to arbitrary error and probe complexity. For example, if you apply a traditional code on an entire database, the resulting codeword can be resistant to error even if a constant fraction of it was corrupted; however, the decoding procedure would typically have to analyze the entire codeword. For large data sizes, this is considered computationally expensive. This may be necessary even if you are only trying to recover a single bit!

The typical process people use is to subdivide their input into sublinear portions, use what is called a "good" code (meaning the encoding is only a constant times larger than the input and the encoding is resistant to a constant fraction of it being corrupted) on each portion, and concatenate the results. But consider what happens. For this codeword, if an adversary were allowed to corrupt just a constant fraction of the codeword, permanent loss of some of the input data could result. This is because the adversary could arbitrarily change one of those sublinear portions and force an error when a decoder tries to retrieve the input bits corresponding to that portion.

LDCs provide the ultimate solution, in some sense. If you know the error probability in advance, you can choose an LDC with the appropriate parameters and protect all your data with very high confidence. We remark that the only type of error LDCs are not resistant against is one that adapts interactively to the decisions of the recovery algorithm – and it is hard to imagine any decoder resistant to that. Moreover, even if you do not know

the error probability in advance, LDCs typically provide graceful degradation in probability of successful recovery for increasing corruption up to a certain limit. If you are able to estimate the amount of corruption present in a data string, you can repeat the recovery algorithm several times to boost the recovery success probability to whatever level you desire.

Katz and Trevisan [32] shows a close correspondence between LDCs and private information retrieval (PIR) schemes, which were introduced in Chor et al. [13]. PIRs are another very practical application. A PIR is a protocol that allows a user to retrieve a bit $i$ of a size $n$ database without the database learning anything at all about $i$. We will define this formally later on. [13] proved that this requires $\Omega(n)$ communication if the database is one server with unlimited computational power. More interestingly, [13] relaxed the PIR problem to allow the database to be many servers which are forbidden from communicating with one another, and showed that this set up allows for PIR protocols with $o(n)$ communication. This gives hope that PIRs could be used in real life, where database sizes can be huge. Examples of particularly useful situations for PIRs include medical and financial databases. In the former, users, who naturally are more likely to query for information on afflictions they have, can get they information they want, while potential eavesdroppers and hackers, who know that users are more likely to query for information on afflictions they have, are unable to figure out what users are querying for, and so are unable to speculate what, if any, affliction a user might have. The

financial case is analogous. Users, who naturally are more likely to query for information of securities they want to buy or sell, can get the information they want, while potential eavesdroppers and hackers, who know that users are more likely to query for information on securities they want to buy or sell, are unable to figure out what users are querying for, and so are unable to speculate what, if any, security a user might want to buy or sell.

Locally testable codes (LTCs) have a related definition to LDCs. Informally, a $q$-LTC is a injective mapping of strings $\mathbf{C}$: $\Sigma^n \rightarrow \Gamma^m$ such that, for any small enough $\delta > 0$, there exists a testing algorithm which outputs "true" when given a string that is an image of $\mathbf{C}$ and which outputs "false" at least half the time when given a string that is at least $\delta m$ Hamming distance from any image of $\mathbf{C}$. Additionally, the algorithm may only examine $q$ of the $m$ positions of the codeword (randomly chosen) in order to decide, and the map $\mathbf{C}$, when viewed as a code, must have $\Omega(m)$ minimum distance.

In the theoretical world, LTCs are valuable in probabilistically checkable proof constructions (for example Goldreich and Sudan [25]). In the practical world, as the definition suggests, using an LTC will allow the receiver of a possibly corrupted data string to quickly check if that data string is close enough to a correct code word. A quick check that returns negative can allow the decoder to save energy by forgoing running a more computationally intensive traditional decoding algorithm.

Self-correcting codes (SCC) also have a related definition to LDCs. Informally, a $q$-SCC is a subset of $\Gamma^m$ such that, for a given $i \in [m]$ and $\delta > 0$, there exists an algorithm which outputs the $i$'th position of the uncorrupted codeword with high probability, even when any set of at most $\delta m$ positions have been corrupted. Furthermore, the algorithm may only examine $q$ of the $m$ positions of the codeword (randomly chosen) in order to decide.

Gertner et al. [23] defines a Symmetric PIR scheme (SPIR) as a PIR scheme which does not information-theoretically leak any data to the user beyond that which he asked for. This may be useful in situations where users have to pay for each access they make to a server. Kalai and Raz [31] uses SPIR to construct small sized non-interactive zero-knowledge proof schemes.

A synonym for an $n$ bit database SPIR scheme used in the literature is a "1-n oblivious transfer scheme." This concept was introduced in Even et al. [19]. There is no difference between these two concepts; however, researchers using the latter terminology are usually less interested in the problem of finding the information-theoretic communication complexity for large $n$. There are many interesting results related to this regarding functions that can computed from several users' inputs, such that those users remain oblivious to certain features of other users' inputs.

Dvir and Shpilka [17] shows a correspondence of LDCs to polynomial identity testing. It relates any depth 3 arithmetic circuit ($\Sigma\Pi\Sigma$ circuit) that computes the zero polynomial to an LDC, and uses a lower bound on the length of LDCs to upper bound the rank of the original circuit.

LDCs have also been used to consider worst case to average case reduction (see Trevisan [43] for an introduction).

Lu et al. [36] uses a variant of LDCs to construct mergers requiring only a small amount of truly random bits to function. With other tricks, this allows them to construct extractors with some of the best known tradeoffs in parameters.

Gasarch [22], Trevisan [43], and Kerenidis and de Wolf [34] detail many other concepts that have similar definitions to LDCs and PIRs. Computationally private information retrieval schemes (CPIR) are PIR schemes that involve an assumption of intractability of some problem. While mathematically they are weaker in the sense that their operation depends on unproven assumptions, they are significantly more efficient in terms of communication and computation than known information theoretic PIR schemes. So they may be more likely to be adopted for use in real life.

## 1.2 The Central Mystery

Katz and Trevisan [32] showed, using an information-theoretic argument, that one query LDCs are essentially impossible. Wehner and de Wolf [44] have the best known lower bound for two query LDCs – it is $2^{\Omega(n)}$, where $n$ is the input size. The best known upper bound for two query LDCs comes from Woodruff [46] and is $2^{O(n)}$. So the known lower and upper bounds for two query LDCs are very close. But the situation is dramatically different for LDCs allowing up to three or more queries! The best lower bound, given by Woodruff [45] is not even larger than $\Omega(n^2)$. On the other hand, Efremenko [18] gives a construction of size $2^{2^{\sqrt{\log n \log\log n}}}$. This was slightly improved by [46] in certain parameter ranges. This dramatic gap has persisted for many years, despite the steadily increasing body of literature on LDCs. From a real world perspective, it is crucial to know whether there exist LDCs with a constant query number that have size polynomial in $n$. If this is true, then LDCs and PIRs are likely to be useful in practice. If this is not true, LDCs and PIRs are much less likely to find adoption – the resources required to implement them for large database sizes would be prohibitive.

# Chapter 2

# General Facts about LDCs

We start with the definition of Locally Decodable Codes, which was given first by Katz and Trevisan in [32]. Our presentation of the definition is based on [32] and Goldreich et al. [24].

**Definition 2.1** *([32]. See also [24].) For reals $\delta$ and $\epsilon$, and a natural number $q$, we say that $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ is a $(q, \delta, \epsilon)$-LOCALLY DECODABLE CODE (LDC) if there exists a probabilistic oracle machine $A$ such that:*

- *In every invocation, $A$ makes at most $q$ queries (possibly adaptively). Query $j \in [m]$ to the oracle $y \in \Gamma^m$ is answered by $y_j$. (Think of $y$ as the potentially corrupted codeword that $A$ is examining.)*
- *For every $x \in \Sigma^n$ and $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$, and for every $i \in [n]$, we have $\Pr[A^y(i) = x_i] \geq \frac{1}{|\Sigma|} + \epsilon$, where the probability is taken over the internal coin tosses of $A$.*

*$A$ is called the DECODING ALGORITHM or DECODER.*

When we study decoding algorithms specifically, sometimes we use the following terminology:

**Definition 2.2** *A probabilistic oracle machine $A$ is said to ACHIEVE $(q, \delta, \epsilon)$ on an LDC $\mathbf{C}$ if*

- *In every invocation, $A$ makes at most $q$ queries (possibly adaptively). Query $j \in [m]$ to the oracle $y \in \Gamma^m$ is answered by $y_j$. (Think of $y$ as the potentially corrupted codeword that $A$ is examining.)*
- *For every $x \in \Sigma^n$ and $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$, and for every $i \in [n]$, we have $\Pr[A^y(i) = x_i] \geq \frac{1}{|\Sigma|} + \epsilon$, where the probability is taken over the internal coin tosses of $A$.*

It is obvious that the following must be true for LDCs to be interesting: $|\Sigma| > 1$, $|\Gamma| > 1$, $0 < \delta \leq 1$, and $0 < \epsilon < 1 - \frac{1}{|\Sigma|}$. For the last parameter range, if instead, $\epsilon = 0$, here is what would happen:

**Fact 2.1** *Any code, for any $q \geq 0$ and any $0 \leq \delta \leq 1$, is a $(q, \delta, 0)$-LDC.*

**Proof:** The recovery algorithm can ignore the output of the code and guess a member of $\Sigma$ uniformly at random. ■

**Definition 2.3** *Let $A$ be an algorithm operating on an LDC $\mathbf{C}$: $\Sigma^n \to \Gamma^m$. Define*

$$\zeta_\delta(A) \triangleq \min_{i \in [n]} \min_{x \in \Sigma^n} \left( \min_{y \in \Gamma^m \,:\, d(y, \mathbf{C}(x)) \leq \delta m} \Pr[A^y(i) = x_i] \right)$$

*(where the probability is over the algorithm's internal randomness) as the algorithm's CORRECTNESS.*

Defining the correctness is convenient especially if one is studying a restricted class of algorithms, some of which have a correctness less than $\frac{1}{|\Sigma|}$. In this case, correctness may be a more intuitive quantity than $\epsilon$.

Here we introduce a common restriction on the capability of recovery algorithms.

**Definition 2.4** *Consider a recovery algorithm A operating on an LDC. A is called NON-ADAPTIVE if A chooses which codeword positions to query without knowledge of the values of any of those positions.*

Very few papers in the literature analyze LDCs that can possibly be adaptive. There are two known reductions from adaptive to non-adaptive codes – they are both in Katz and Trevisan [32]. The first only changes $q$ but leaves $\delta$ and $\epsilon$ the same. The second only changes $\epsilon$, but leaves $q$ and $\delta$ the same. The only other paper dealing with possibly adaptive codes is Deshpande et al. [15], which proves a lower bound very similar to [32] for them. Not surprisingly, the proof techniques in [15] are substantially more complicated. In this document, we only study non-adaptive algorithms. This motivates the following definition:

**Definition 2.5** *Consider a non-adaptive recovery algorithm A operating on an LDC. Assume A has been tasked to return the i'th ($i \in [n]$) input bit of the code. Then without loss of generality, A can flip all of its internal random*

*coins before doing any other operation. For some specific value of internal randomness, define the QUERY SET as the values $j \in [m]$ representing the codeword positions A has chosen to query.*

Clearly if, for a given $i$ and a given value of internal randomness, the size of the query set is less than $q$, we could add additional vertices to the query set so that it has size exactly $q$ – the algorithm can just ignore the results of the extra queries. In general, when we refer to a $q$ query algorithm, that algorithm is allowed to query less than $q$ queries sometimes. But when we say an EXACTLY $q$ query algorithm, we mean an algorithm that always queries and uses exactly $q$ positions.

**Definition 2.6** *A code $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ is called BINARY if $|\Sigma| = |\Gamma| = 2$.*

It is often simpler to prove results for binary codes than for more general codes, so some of our results will hold for binary codes only.

**Definition 2.7** *Take a field $F$. A function $\mathbf{C}$: $F^n \to F^m$ is called a LINEAR MAPPING if, for all $a, b \in F^n$, $\mathbf{C}(a) + \mathbf{C}(b) = \mathbf{C}(a + b)$.*

The following definition is closely related.

**Definition 2.8** *Take a field $F$. A linear subspace of $F^m$ is a called a LINEAR CODE.*

Linear codes are widely studied by researchers because they often have nice combinatorial structure. They are also used widely in practical implementations because encoding is simple.

For a linear code, it is convenient to represent the function that determines a given codeword position by a vector: for a given $j \in [m]$, define $a_j \in F^n$ as the $a_j$ satisfying $\forall x \in F^n$, $\mathbf{C}_j(x) = a_j \cdot x$.

Here is notation we will use for analyzing linear codes:

**Definition 2.9** *Let $e_i$ denote the i'th unit vector with length $n$.*

For a given edge $Q$, we use the notation span($Q$) to represent the linear span of the vectors in $Q$.

We make extensive use of probability expressions in this thesis. To make notation compact when conditioning on long expressions, for two events $A$ and $B$, $A;B$ will denote the event $A \cap B$.

Here is a simple observation that we will occasionally use:

**Claim 2.2** *Let $\mathbf{C}$ be a $(q, \delta, \epsilon)$-LDC of length $m$. Then there also exists an $(q, \delta, \epsilon)$-LDC with no codeword position that is identically a constant value and which has length at most $m$.*

**Proof:** Let $A$ be an algorithm for $\mathbf{C}$ achieving $(q, \delta, \epsilon)$. This of course means that:

$$\min_{i\in[n]} \min_{x\in\Sigma^n} \left( \min_{y\in\Gamma^m \,:\, d(y,\mathbf{C}(x))\leq\delta m} \Pr[A^y(i) = x_i] \right) \geq \frac{1}{2} + \epsilon$$

$$\Rightarrow \min_{i\in[n]} \min_{x\in\Sigma^n} \left( \min_{y\in\Gamma^m \,:\, d(y,\mathbf{C}(x))\leq\delta m; y=\mathbf{C}(x) \text{ on constant positions}} \Pr[A^y(i) = x_i] \right) \geq \frac{1}{2} + \epsilon$$

Now let us construct a new algorithm $A'$ that is the same as $A$ except for the following. Whenever $A$ would have queried a constant position, $A'$ uses the known, uncorrupted value, rather than using the results of the query. For $A'$, it is also the case that:

$$\min_{i\in[n]} \min_{x\in\Sigma^n} \left( \min_{y\in\Gamma^m \,:\, d(y,\mathbf{C}(x))\leq\delta m; y=\mathbf{C}(x) \text{ on constant positions}} \Pr[A'^y(i) = x_i] \right) \geq \frac{1}{2} + \epsilon$$

Now construct a code **C'** that is **C** excluding the constant positions. Call the length of **C'** $m'$. For **C'** and $A'$ it is still the case that:

$$\min_{i\in[n]} \min_{x\in\Sigma^n} \left( \min_{y\in\Gamma^{m'} \,:\, d(y,\mathbf{C}'(x))\leq\delta m} \Pr[A'^y(i) = x_i] \right) \geq \frac{1}{2} + \epsilon$$

$$\Rightarrow \min_{i\in[n]} \min_{x\in\Sigma^n} \left( \min_{y\in\Gamma^{m'} \,:\, d(y,\mathbf{C}'(x))\leq\delta m'} \Pr[A'^y(i) = x_i] \right) \geq \frac{1}{2} + \epsilon \qquad \text{because } m' \leq m$$

Thus, **C'** is a $(q, \delta, \epsilon)$-LDC. ■

## 2.1 Smooth Codes

In this section, we discuss smooth codes, which have great similarity to LDCs. We start by defining what a smooth code is. The definition was given

first by Katz and Trevisan in [32]. Our presentation of the definition is based on [32] and Goldreich et al. [24].

**Definition 2.10** *([32]. See also [24].) For a natural number $q$ and positive reals $c$ and $\epsilon$, we say that $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ is a $(q, c, \epsilon)$-SMOOTH CODE if there exists a probabilistic algorithm $A$ such that:*

- *In every invocation, $A$ reads at most $q$ positions of $\mathbf{C}(x)$.*
- *For every $i \in [n]$ and $x \in \Sigma^n$, we have $\Pr[A^{\mathbf{C}(x)}(i) = x_i] \geq \frac{1}{|\Sigma|} + \epsilon$.*
- *For every $i \in [n]$ and $j \in [m]$, the probability that on input $i$, $A$ queries index $j$ is at most $\frac{c}{m}$.*

*The probabilities are taken over the internal coin tosses of $A$. Just as for LDCs, $A$ is called the DECODING ALGORITHM.*

[32] gives reductions of smooth codes to and from LDCs:

**Lemma 2.1.1** *([32].) A $(q, c, \epsilon)$-smooth code is also a $(q, \delta, \epsilon - c\delta)$-LDC.*

**Lemma 2.1.2** *([32].) A $(q, \delta, \epsilon)$-LDC is also a $(q, \frac{q}{\delta}, \epsilon)$-smooth code.*

As in the case of LDCs, $A$ is called non-adaptive when $A$'s choice of what to query does not depend on its previous queries. The correctness of a smooth code algorithm is defined just the same as the correctness of an LDC

algorithm.

Related to smooth codes are smooth decoders of codes. We will now analyze some comments given by Katz and Trevisan [32] [43] regarding LDCs whose recovery algorithms are restricted to "smoothly" distribute their queries across the codeword positions:

**Definition 2.11** *([43]. See also [32].) Let* $\mathbf{C}$*:* $\Sigma^n \to \Gamma^m$ *be a code. Assume there exists a probabilistic algorithm* $A$ *such that:*

- *For every* $i \in [n]$ *and* $x \in \Sigma^n$*, we have* $\Pr[A^{\mathbf{C}(x)}(i) = x_i] \geq \frac{1}{|\Sigma|} + \epsilon$.
- *For every* $i \in [n]$ *and* $j \in [m]$*, the probability that on input* $i$*,* $A$ *queries index* $j$ *is at most* $\frac{c}{m}$.

*The probabilities are taken over the internal coin tosses of* $A$*. Then we call* $A$ *a* $(c, \epsilon)$*-SMOOTH DECODER.*

*If, in addition, for every* $i \in [n]$ *and* $j_1, j_2 \in [m]$*, the probabilities that on input* $i$*,* $A$ *queries index* $j_1$ *or* $j_2$ *are exactly the same, then* $A$ *is called a PERFECTLY SMOOTH DECODER.*

Note that given a length $m$ code, any decoding algorithm is an $(m, \epsilon)$-smooth decoder for some $\epsilon$. Also, any local decoding algorithm that always probes exactly $q$ queries cannot be a $(c, \epsilon)$-smooth decoder for $c < q$. Otherwise, the sum of the probabilities of querying each codeword position would be

less than $q$, contradicting the assumption that the decoder always probes exactly $q$ positions. By similar logic, a $(q, \epsilon)$-smooth decoder that always probes exactly $q$ queries is a perfectly smooth decoder. No codeword position can be queried with probability less than $\frac{q}{m}$ because otherwise, the sum of the probabilities of querying each codeword position would be less than $q$, contradicting the assumption that the decoder always probes exactly $q$ positions. So all positions are queried with the same probability: $\frac{q}{m}$.

With the definition of smooth decoder in hand, we can prove:

**Claim 2.1.3** *([32]. See also [43].) Let* $\mathbf{C}$*:* $\Sigma^n \to \Gamma^m$ *be a code. Assume that an algorithm* $A$ *is a* $(c, \epsilon)$*-smooth decoder for* $\mathbf{C}$*. Then* $\zeta_\delta(A) \geq \frac{1}{|\Sigma|} + \epsilon - c\delta$*.*

**Proof:** Fix $i \in [n]$ as the input position the algorithm is tasked to recover. Let $y$ be $\mathbf{C}(x)$ corrupted in at most $\delta m$ positions by the adversary. The probability that any one of the $\delta m$ corrupted positions is queried by $A^y(i)$ is at most $\frac{c}{m}$. By the union bound, the probability of the algorithm querying any of the corrupted positions is at most $\frac{c}{m}\delta m = c\delta$. Also, by assumption, the algorithm is wrong with probability at most $1 - \frac{1}{|\Sigma|} - \epsilon$ when operating on $\mathbf{C}(x)$ uncorrupted. So, using another union bound, the probability the algorithm is wrong on $y$ is at most $1 - \frac{1}{|\Sigma|} - \epsilon + c\delta$. ■

## 2.2 Overview of Previous Results

In this section, we give a broad overview of the results people have proven for different parameter ranges of LDCs. In the sections that follow, we will delve into detail about how the results fit into the context of history.

Katz and Trevisan [32] proved, using an information-theoretic argument, that for $q = 1$, $\epsilon > 0$, and large enough $n$, LDCs do not exist. Wehner and de Wolf [44] have the best lower bound for $q = 2$ LDCs: assuming $|\Sigma| = 2$ and $|\Gamma| = 2^l$, the length of the code must be at least $2^{\Omega(\frac{n}{2^{2l}})}$. (This lower bound and all of the lower bounds below assume $\epsilon > 0$.) If we can assume that the recovery procedure uses only $b$ bits from each codeword position it queries, their lower bound improves to $2^{\Omega(\frac{n}{2^b \sum_{i=0}^{b} \binom{l}{i}})}$. If we can assume the LDC is binary and linear, Shiowattana and Lokam [42] prove a lower bound of $\Omega(2^{\frac{4\delta n}{1-2\epsilon}})$, and that is tight within a constant factor. The best known binary, $q = 2$ LDC is the Trevisan construction, detailed by Obata [38]. It has size $O(2^{\frac{4\delta n}{1-2\epsilon}})$. It exists for any tradeoff of $\delta$ and $\epsilon$ so long as $\epsilon \leq \frac{1}{2} - 2\delta$. Actually, Woodruff [46] provides a transformation that for small enough $\epsilon$ and $\delta$ makes the Trevisan construction even shorter. This transformation is interesting because it makes the code non-linear, in contrast to almost all other constructions in the literature which are linear. The size becomes $2^{O(\max(\delta,\epsilon)\delta n)}$. So the known upper and lower bounds for $q = 2$, binary LDCs are very close, and, under the assumption of linearity, even closer. But the situation is dramatically different for $q \geq 3$. The best lower bound for $q \geq 3$, given by Woodruff [45], is $\Omega(\frac{n^{\frac{q+1}{q-1}}}{\log n})$.

If we can assume the LDC is linear and $q = 3$, Woodruff again in [47] proves the best lower bound of $\Omega(n^2)$. The best upper bound for $q \geq 3$, given by Efremenko in [18], is $2^{2^{O(\sqrt{\log n \log\log n})}}$. This construction has a non-binary version with correctness $1-3\delta$ and a binary version with correctness $1-6\delta$. For small enough $\epsilon$ and $\delta$, LDCs can be improved slightly by a black box transformation mentioned in [46]. Also, there is another transformation in [46] that converts the binary Efremenko code into one with correctness $1-3\delta-\alpha$, for any $\alpha > 0$, at the expense of blowing up the code's length.

Two thorough surveys of LDCs are Yekhanin [50] and Hielscher [26].

## 2.3 Previous LDC Impossibility Results

Katz and Trevisan [32] showed, using an information-theoretic argument, that for $q = 1$, $\epsilon > 0$, and large enough $n$, LDCs do not exist. For $q = 2$ LDCs, the existence of the Hadamard code rules out a general impossibility result, but we ask the question of what is the allowed tradeoff between $\epsilon$ and $\delta$. We will prove later that when $q = 2$, linear LDCs can only exist for $\epsilon \leq \max(0, \frac{1}{2} - \frac{2\delta n}{n+1})$.

## 2.4 Previous LDC Lower Bounds Results

After Katz and Trevisan [32] but before Kerenidis and de Wolf [34], the huge gap between known upper and lower bounds started at $q = 2$. To try to bridge that gap, a series of papers beginning with Goldreich et al. [24], followed by Obata [38], and continuing most recently with Shiowattana and Lokam [42], and Dvir and Shpilka [17] considered only linear, $q = 2$ LDCs. [24] proved that, for large enough $n$, binary, linear, $q = 2$ LDCs must have size at least $2^{\epsilon\delta n/4}$. (This lower bound and all of the lower bounds below assume $\epsilon > 0$.) They also proved that linear, $q = 2$ LDCs over a finite field $F$ must have size at least $\frac{2^{\epsilon\delta n/8}}{4|F|}$. [38] improved the exponent: he showed that, for large enough $n$, binary, linear, $q = 2$ LDCs must have size at least $2^{\frac{1}{4.03}\frac{\delta}{1-2\epsilon}n}$. [42] improved this further: they showed that binary, linear, $q = 2$ LDCs must have size at least $2^{\frac{4\delta}{1-2\epsilon}n-1}$. This is very impressive as it is within a constant factor of the best known upper bound of this type of code, the Trevisan construction. [17] improved the lower bound for non-binary codes: they showed that linear, $q = 2$ LDCs over any (possibly infinite) field must have size at least $2^{(\epsilon\delta n/4)-1}$.

The result of Kerenidis and de Wolf [34] was shocking because it proved strong lower bounds on the size of possibly non-linear LDCs using quantum proof techniques, even though LDCs seemingly have nothing to do with quantum computing. Specifically, [34] proved that $q = 2$ LDCs with $\Sigma = \{0, 1\}$ and $\Gamma = \{0, 1\}^l$ must have size at least $2^{H(1/2+\delta\epsilon/2^{3l+1})n-l}$. Wehner and de Wolf [44] improve this slightly: they show that $q = 2$ LDCs with $\Sigma = \{0, 1\}$

and $\Gamma = \{0,1\}^l$ whose recovery algorithms only use $b$ bits of the $l$ they get in each query, must have size at least $2^{\Theta(\delta\epsilon^2/(2^b u))n - log(u)}$ with $u = \sum_{i=0}^{b} \binom{l}{i}$. Many people, including Jain [30] and Samorodnitsky [41], had asked whether quantum techniques are essential or if they are just an artifact of how the proof was discovered. [30] proves a result for smooth codes instead of LDCs, but because there exist reductions of smooth codes to and from LDCs [32], that is considered very relevant. [30] shows that binary, $q = 2$ smooth codes with $\frac{1}{2} + \epsilon \geq 1 - \frac{c^2}{8n^2}$ must have size at least $2^{\frac{n}{320c^2}-1}$. [30]'s proof does not use quantum techniques but has a much worse bound that holds for a much narrower range of parameters. [41] wrote many of the quantum steps of [44] in non-quantum mathematics. Ben-Aroya et al. [8] finally proved an exponential lower bound on the size of two query, binary, possibly non-linear LDCs without the use of quantum techniques. They used a deep mathematical inequality that they showed had many other applications. [34] shows how an LDC reduction credited to Trevisan can transform binary LDC lower bounds into lower bounds on codes with somewhat higher output alphabet size. This can be used on both [34]'s and [8]'s lower bounds. However, a non-trivial lower bound on the size of two query, possibly non-linear LDCs over larger fields is not known, except when the input alphabet is binary and the output alphabet size is less than $O(\sqrt{n})$.

Despite the small gap between upper and lower bounds for $q = 2$, there is still a big gap for $q \geq 3$. The first lower bound on $q \geq 3$ LDCs

came from [32]. [32] showed that LDCs with $\Sigma = \{0, 1\}$ must have size at least $\Omega((\frac{n}{\log |\Gamma|})^{\frac{q}{q-1}})$. Deshpande et al. [15] prove the same lower bound even when the LDC's recovery algorithm may be adaptive. Not surprisingly, considering adaptive algorithms makes their proof much more complicated. [44] improves [32] slightly: they show that binary LDCs must have size at least $\Omega((\frac{n}{\log n})^{1+1/(\lceil q/2 \rceil - 1)})$. The bound was improved slightly more by Woodruff [45]: he proved that odd $q \geq 3$ query, binary LDCs must have size at least $\Omega(\frac{n^{\frac{q+1}{q-1}}}{\log n})$. He later proved [47] that linear, three query LDCs over any (possibly infinite) field, must have size at least $\Omega(n^2)$. It is important to note that the $q \geq 3$ results of both [44] and [45] are fundamentally based on the same combinatorial theorem of [32]. The combinatorial part of [15] is similar to that of [32] as well.

(As a footnote to the above discussion on quantum techniques in analyzing LDCs, Briet and de Wolf [11] studied a different object they called Locally Decodable Quantum Codes (LDQC). An LDQC is defined as a mapping $\Sigma^n$ to $m$ qubits such that, for each $i \in [n]$, the symbol $x_i$ can be recovered with probability at least $\frac{1}{|\Sigma|} + \epsilon$ by making only $q$ quantum queries, even if $\delta m$ qubits of the codeword have been corrupted. [11] concluded "that $q$-query quantum codes are not significantly better than $q$-query classical codes, at least for constant or small $q$." Thus, more research has been focussed on quantum techniques in analyzing LDCs rather than on quantum analogs of LDCs.)

## 2.5 Previous LDC Constructions

The prototypical LDC construction is the Hadamard code. This is an old construction, but it was first studied in a similar way to LDCs by Blum et al. [10]. The length of this code is $2^n$. Many constructions of LDCs were PIR constructions that were adapted into LDCs. These will be discussed in the next section. Trevisan gave the first length improvement on the Hadamard code that did not originally come from studying PIR. His construction has length $O(2^{\frac{4\delta n}{1-2\epsilon}})$. Woodruff [46] made a general transformation that, for small enough $\delta$ and $\epsilon$, converts this linear code into a non-linear code of size $2^{O(\max(\delta,\epsilon)\delta n)}$. Interestingly, this code is smaller than the best lower bound for linear $q = 2$ codes. For $q \geq 3$, the best known construction, given by Efremenko [18], is $2^{2^{O(\sqrt{\log n \log\log n})}}$. This builds on a series of papers originating from the breakthrough result of Yekhanin [49]. All of these LDCs have parameters $\frac{1}{2} + \epsilon = 1 - 6\delta$. [49]'s construction of length $O(\exp(n^{10^{-7}}))$ was significantly smaller than the previous record holder, Beimel et al. [6] and its bound of $\exp(\exp(O(\frac{\log n \log^{(2)} q}{q \log q})))$. [49] was also unique because it showed, if one could assume a widely believed assumption about Mersenne primes (specifically, that there are infinitely many of them), a construction of length $O(\exp(n^{O(1/\log\log n)}))$ was possible. Raghavendra [39] simplified [49]'s argument. Kedlaya and Yekhanin [33] slightly relaxed the assumption needed to create these short codes. They only required the existence of an infinite number of Mersenne numbers with large prime factors. In any event, [18] supersedes all of the preceding results, because it has smaller length, and it does

not use any unproven assumptions. As stated earlier, there are two transformations given by Woodruff [46] that can operate on the binary Efremenko code: one converts the code into one with correctness $1-3\delta-\alpha$, for any $\alpha > 0$, at the expense of blowing up the code's length. The other, for small enough $\epsilon$ and $\delta$, reduces the length of the code slightly.

For larger numbers of queries, the above results achieve correctness that decreases with increasing $q$, of the form $1-q\delta$, and thus they can tolerate only small $\delta$ for large $q$. In fact, our results will show that with the type of decoders they use, they cannot possibly do better. Dvir et al. [16] achieve better dependence of the correctness on the corruption as $q$ increases, and provide binary constructions that can tolerate close to $1/8$ fraction of corruption for large numbers of queries. Ben-Aroya et al. [7] give a sub-exponential size construction that tolerates up to $\frac{1}{2}-\alpha$ fraction of corruption, however the field size and number of queries blow up as $\alpha$ decreases to 0. Babai et al. [2] give a construction of size $n^{1+\alpha}$, for any $\alpha > 0$, which requires a poly-logarithmic number of queries. If $n^{\alpha}$ queries are allowed, then Kopparty et al. [35] show that the rate of the LDC can be arbitrarily close to 1. We will see that a significant innovation of these papers is that the decoding algorithms presented in them use non-linear operations to enhance their tolerance to codeword corruption.

## 2.6 Previous PIR Results

Katz and Trevisan [32] prove reductions of LDCs to and from smooth codes and reductions of smooth codes to and from PIRs. So the study of LDCs and PIRs are intertwined. We start with the formal definition of PIR:

**Definition 2.12** *(Chor et al. [13].) A one-round, k-server PRIVATE INFORMATION RETRIEVAL (PIR) scheme for a database of length n consists of:*

- *k query functions:* $Q_1, ..., Q_k : [n] \times \{0,1\}^{l_r} \to \{0,1\}^{l_q}$
- *k answer functions:* $A_1, ..., A_k : \{0,1\}^n \times \{0,1\}^{l_q} \to \{0,1\}^{l_a}$
- *One reconstruction function:* $R : [n] \times \{0,1\}^{l_r} \times (\{0,1\}^{l_a})^k \to \{0,1\}$

  *These functions should satisfy:*

- *Correctness: For every* $x \in \{0,1\}^n$, $i \in [n]$, *and* $r \in \{0,1\}^{l_r}$,

$$\Pr[R(i, r, A_1(x, Q_1(i, r)), ..., A_k(x, Q_k(i, r))) = x_i] \geq \frac{1}{2} + \epsilon$$

- *Privacy: For every* $i, j \in [n]$, $s \in [k]$, *and* $q \in \{0,1\}^{l_q}$,

$$\Pr[Q_s(i, r) = q] = \Pr[Q_s(j, r) = q]$$

  *where the probabilities are taken over uniformly random* $r \in \{0,1\}^{l_r}$. $\frac{1}{2} + \epsilon$ *is the probability of correct retrieval –* $\epsilon$ *is often required to be* $\frac{1}{2}$. $l_q$ *is called the query size, and* $l_a$ *is called the answer size.*

There are many reasonable extensions of this definition (which we will not consider). For example, the protocol could utilize more than one round of user/server communication. Or the privacy constraint could be weakened from information theoretic security to computational security.

Here are the mentioned reductions of PIRs to and from smooth codes:

**Lemma 2.6.1** *([32].) A PIR scheme with $k$ servers, query size $l_q$, answer size $l_a$, and probability of correct retrieval $\frac{1}{2}+\epsilon$ gives a $(k, k, \frac{\epsilon}{2})$-smooth code $\mathbf{C}$: $\{0,1\}^n \to (\{0,1\}^{l_a})^m$ with $m = O(\frac{k2^{l_q}}{\epsilon})$.*

**Lemma 2.6.2** *([32].) A $(q, c, \epsilon)$-smooth code $\mathbf{C}$: $\{0,1\}^n \to \Sigma^m$ gives a PIR scheme with $q$ servers, query size $\log m$, answer size $\log |\Sigma|$, and probability of correct retrieval $\frac{1}{2}+\frac{\epsilon^2}{2c}$.*

Now we describe some previous results on PIR.

Mann [37] proves that for a $k \geq 2$ server PIR scheme operating on a database of size $n$ bits such that each each server is sent the same number of bits from the user, the total communication must be at least $(\frac{k^2}{k-1} - \alpha) \log n$ for any $\alpha > 0$. This is interesting because for small enough $\alpha$ and large enough $n$, it is strictly bigger than the amount of communication required for a user to retrieve an item from a database without insisting on privacy (i.e. $\log n + 1$). Itoh [29] has strong lower bounds, but it assumes the servers represent their

data in unusual ways that rule out many known constructions. Beigel et al. [3] proved that any one round, two server PIR scheme in which the servers always return one bit must have queries of size at least $n - 2$. This is a strong lower bound, but they only consider a very restricted class of PIRs. Razborov and Yekhanin [40] proved a $\Omega(n^{1/3})$ lower bound on the communication complexity of what are called bilinear, one round, two server PIR schemes. Bilinear schemes, which encompass all known two server PIR schemes except those coming from a transformation by Woodruff [46], are those in which the user takes a generalized dot product of the servers' response vectors in order to calculate the desired database bit. This is a great result because the bounds discussed earlier for LDCs do not translate well into this type of PIR – the best general $q = 2$ LDC lower bound, coming from Wehner and de Wolf [44], implies a lower bound of $(5 - o(1)) \log n$ on the communication complexity of one round, two server PIR schemes. Additionally, this lower bound matches the communication complexity of the best known construction for a one round, two server PIR scheme (see below), which is $O(n^{1/3})$.

As for upper bounds, Chor et al. [13] show the existence of a two server PIR scheme with total communication complexity $O(n^{1/3})$. This was generalized by Ambainis [1], who showed the existence of $k \geq 2$ server PIR schemes with total communication complexity $O(2^{k^2} n^{1/(2k-1)})$. Itoh [28] improved the constant in front to get $O(k!\ n^{1/(2k-1)})$. Beimel, Ishai, and Kushilevitz [27] [4] [5] improved that constant further in new constructions to $O(k^3 n^{1/(2k-1)})$.

These papers produced more significant improvements for what is called $t$-private PIR, which is when no coalition of $t \leq k$ servers can determine the user's query. (Regular PIR can be considered as 1-private PIR.) Finally, Beimel et al. [6] construct $k \geq 3$ server PIR schemes with total communication complexity $n^{O(\frac{\log\log k}{k \log k})}$, beating the $O(n^{1/(2k-1)})$ roadblock from before. One of their constructions can be converted into a binary, $q$ query LDC of length $2^{n^{O(\log\log q/(q \log q))}}$. Woodruff and Yekhanin [48] produced another construction of size $O(k^2 \log k n^{1/(2k-1)})$ that had improvements for $k = 2$ PIR and $t$-private PIR with $t > 1$. After this, a line of work starting with Yekhanin [49] created dramatic size reductions for LDC constructions directly instead of PIR constructions.

These constructions are families that, for higher $k$, give $k$-server PIR schemes that have smaller and smaller communication complexity. Therefore, when these PIR schemes are converted into LDCs, they give $q$ query LDCs that are shorter for higher $q$.

Chor et al. in [13] give a scheme with $k = \log n$ servers and $\exp((\log k)^2 \log\log k)$ communication complexity. This is one of the only papers to address PIRs that allow a higher than constant number of servers.

Gasarch has a survey of PIR [22] that covers similar material to this chapter, but it goes into more depth.

# Chapter 3

# Main Results Completed

In this chapter, we give an overview of the most important results of this dissertation.

## Basic Impossibility Results

We prove several impossibility results for LDCs. An impossibility result for one query LDCs was given by Katz and Trevisan [32]. They show that one query LDCs with binary inputs do not exist for any $\delta$ and $\epsilon$, for large enough input sizes and small enough output alphabet sizes. We generalize this to any input alphabet. In addition, we prove a few basic impossibility results that hold without restriction on how many queries the recovery algorithms are allowed to make. First, we show that if any recovery algorithm, operating on any binary code, performs better than random guessing, then it must be the case that $\delta \leq \frac{1}{2}$ (Claim 4.1). We extend this to codes over any input alphabet: if any recovery algorithm, operating on any code, performs better than random guessing, then it must be the case that $\delta \leq 1 - \frac{1}{|\Sigma|}$ (Claim 4.2). Thus, we prove there is a range of $\delta$ for which no LDC with $\epsilon > 0$ can exist. Next, we relate the ability of a recovery algorithm to find the requested input bit better than

random guessing with the underlying code's minimum distance. We prove that the minimum distance for binary codes is at least $2\delta m + 1$ (Lemma 4.3). This also implies Claim 4.1. For linear codes, we can extend this to higher input alphabet sizes: if the input alphabet is the field $F$, the minimum distance is $\frac{|F|}{|F|-1}\delta m + 1$ (Lemma 4.4).

## Length Lower Bounds

We study how the length of a code influences the correctness of algorithms operating on it. Describing this another way, we show how the existence of algorithms with very high correctness operating on a specific code implies something about the length of that code. Our result is the first large length lower bound for LDCs that are allowed to use more than two queries. Our main results show that achieving slightly larger than $1 - 3\delta$ correctness for three query locally decodable codes requires exponential length. Subexponential length constructions exist for three query LDCs up to $1-3\delta$ correctness for codes over large, non-constant size fields [18]. No three query, binary LDC of sub-exponential size has been demonstrated to achieve correctness of $1-3\delta$ or more (for $\delta < \frac{1}{6}$). For binary codes, Woodruff [46] demonstrates three query, binary, linear LDCs with correctness $1 - 3\delta - \eta$, where $\eta > 0$ is an arbitrarily small constant, having sub-exponential length. Our lower bound is $\exp(\Omega(n))$ and holds for LDCs with correctness at least $1 - 3\delta + 6\delta^2 - 4\delta^3 + O(\frac{1}{n^{1/3}})$. Thus, our lower bound shows that achieving slightly larger correctness than the currently known subexponential length constructions already requires ex-

ponential length.

Our results show, surprisingly, that the smallest three query linear LDCs have significantly different structures in different parameter ranges. Contrast this with two query linear codes, where there are almost matching lower [42] and upper bounds [38] for the code length ($\Omega(2^{\frac{4\delta}{1-2\epsilon}n})$) that hold regardless of the LDC's correctness, as long as $\epsilon > 0$ is an arbitrary constant.

We have generalized this lower bound to three query, binary codes (Theorem 6.2.2), to three query, linear codes over arbitrary finite fields (Theorem 6.3.5), and to four query, binary, linear codes (Theorem 7.1). Each of these proofs uses somewhat different methods. The four query lower bound handles an interesting case where two bits of a query are the same function of the input. This is interesting, because one might intuitively believe that having multiple copies of the same input bit could lead to a better performing LDC (higher correctness or smaller length for a given corruption level). But the bound we obtain for four query, binary, linear codes is extremely close to the bound we obtain for three query, binary, linear codes. Our bound shows that adding any fourth bit does not seem to help much.

Additionally, we extended our length lower bounds to arbitrary $q \geq 3$, assuming that the recovery algorithm is of one of three commonly used types. We start by considering linear decoders. A linear decoder is a recovery algo-

rithm that returns a fixed linear combination of the positions it reads. If a linear decoder achieves correctness $1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + O(\frac{1}{n^{1/3}})$ for a binary or linear code with $\delta$ fraction corruption, then the length of the code must be exponential (Theorem 8.1).

Matching sum decoders were formally defined by Woodruff [46]. A $q$ query matching sum decoder picks a set of size $q$ uniformly at random from a collection of sets that form a matching in the complete $q$ uniform hypergraph, whose vertices correspond to the positions of the codeword. Then, the decoder reads the positions corresponding to the chosen set, and returns the sum of the positions read. Many known constructions of locally decodable codes have such decoders. If a matching sum decoder achieves correctness $1-3\delta+O(\frac{1}{n^{1/3}})$ for a binary or linear code with $\delta$ fraction corruption, then the length of the code must be exponential (Theorem 8.2).

Consider a recovery algorithm operating on a linear code that only queries query sets that are linearly independent. If, for a given $\delta$ fraction codeword corruption, the recovery algorithm achieves correctness $\frac{1}{|F|} + \epsilon$, then the length of the code must be at least exponential in the quantity $n(\delta + \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3} + \frac{1}{|F|} - 1 - (\frac{108|F|}{n})^{1/3}) - O(1)$ (Theorem 8.4).

## Bounds on Correctness

In the Impossibility Results section, we described results that show that for certain values of $\delta$, no LDCs with $\epsilon > 0$ exist at all. Now we consider the largest $\epsilon$ achievable for a given $\delta$. Goldreich et al. [24] notes the following: "Note that in a locally decodable code that corrects up to a $\delta$ fraction of errors, the reconstruction probability cannot be arbitrarily close to 1." However, we did not find any bounds along these lines in the literature. We believe our bounds on correctness fill an important gap in the literature.

First, we prove upper bounds on the correctness of recovery algorithms under restrictions on their behavior. These restrictions we make are very common in constructions in the literature. We show that any linear decoder has correctness at most $1 - 2\delta + \frac{|F|}{|F|-1}\delta^2 + O(\frac{1}{\sqrt{n}})$ (Lemma 5.3.3). The concept of a linear decoder has been implicit since the first paper studying linear LDCs [24]. We discuss this in more details later. We are also able to prove that the correctness of a matching sum decoder is no more than $1 - q\delta$ (Claim 5.3.4). The last special type of algorithms we consider is that which only operates on linear codes and only queries codeword positions that are linearly independent. We show that they cannot achieve correctness more than $1 - q\delta + o(\delta) + O(\frac{1}{n})$ (Claim 8.3).

We then advance to proving upper bounds on the correctness of arbitrary recovery algorithms, without restrictions. We prove a neat linear algebra

property that allows us to upper bound the correctness of $q$ query, binary, linear codes below $1-2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 \rfloor -1}\delta^{\lceil q/4 \rceil +1}(1-\delta)^{\lfloor q/4 \rfloor}+O(\frac{1}{n})$ (Theorem 9.2.1). Later on, we provide a construction (Theorem 12.2) showing that this bound is tight. Then we generalize the linear algebra property, using a probabilistic method. This allows us to upper bound the correctness of $q$ query, binary, possibly nonlinear codes as at most roughly $1-1.98\frac{\delta}{\sqrt{q}}\Big(4\delta(1-\delta)\Big)^{\frac{q}{4}}$ for large enough $n$ (see Theorem 9.4.1 for the precise statement). Finally, for any algorithm operating over a possibly nonlinear code with alphabet $\Sigma$, we prove the correctness is at most $1-2\frac{|\Sigma|-1}{|\Sigma|}\binom{q}{\lfloor (q-1)/2 \rfloor}(\frac{\delta}{|\Sigma|-1})^{\lceil (q+1)/2 \rceil}(1-\delta)^{\lfloor (q-1)/2 \rfloor}+O(\frac{1}{n})$ (Theorem 9.5.1).

The bounds just discussed are most interesting for large $q$ – and, in fact, in the binary, linear case, our bound is close to tight in some sense. For small $q$, these bounds are not tight. To address this, we also prove precise bounds on the correctness of recovery algorithms that make no more than two or three queries. We prove that the correctness of two query recovery algorithms working on binary, linear codes is at most $\max(\frac{1}{2}, 1-\frac{2\delta n}{n+1})$ (Claim 10.1). Note that correctness $\max(\frac{1}{2}, 1-2\delta)$ is achievable by the Hadamard code (Lemma 12.1), which is linear. We proceed with an upper bound on the correctness of three query recovery algorithms working on binary, linear codes, proving the correctness is at most $1-2\delta(1-\delta)+O(\frac{1}{n})$ (Claim 10.2). Claim 10.3 shows that two query decoders working on binary codes have correctness at most $1-2\delta(1-\delta)+O(\frac{1}{n^{1/3}})$. The methodology of these proofs turned out to be very helpful in proving our length lower bounds described earlier.

## Locally Decodable Erasure Codes

Locally Decodable Erasure Codes (LDECs) are intimately related to LDCs. More concretely, LDECs are LDCs but they must tolerate codeword erasures instead of codeword corruptions. We initiate a study of their properties by comparing them to LDCs. We also prove several correctness bounds about them.

## Hadamard Codes

The Hadamard code seems to be the prototypical example of an LDC. Moreover, the Hadamard code may achieve the highest $\epsilon$ for any $\delta$ and $q$. For this reason, we have precisely analyzed the performance of a good $q$ query decoder operating on the Hadamard code: the error is at most $\Big(4(2\delta)(1-2\delta)\Big)^{\lfloor \frac{q}{4} \rfloor}$ (Theorem 12.2). These results show that the Hadamard code can indeed achieve close to optimal correctness.

## Note

Some of these results appear in Gál and Mills [20] [21]. Our bounds on correctness will appear in an upcoming paper by Cheraghchi, Gál, and Mills [12].

# Chapter 4

# Basic Impossibility Results

In this chapter, we start exploring the limits of correctness of LDC recovery algorithms operating on codes. These impossibility results show that, for certain $\delta$, LDCs with correctness better than random guessing do not exist. The definition of the correctness of a recovery algorithm has a minimum over all input positions the algorithm could be tasked to recover. When our proofs are sufficiently general that they work for all input positions, we emphasize that by considering the following variant of correctness:

**Definition 4.1** *Let $A$ be an algorithm operating on an LDC $\mathbf{C}$: $\Sigma^n \to \Gamma^m$. Define*

$$\zeta^\star_\delta(A) \triangleq \max_{i\in[n]} \min_{x\in\Sigma^n} \left( \min_{y\in\Gamma^m \,:\, d(y,\mathbf{C}(x))\le\delta m} \Pr[A^y(i) = x_i] \right)$$

*where the probability is over the algorithm's internal randomness.*

We start by giving bounds that hold for all algorithms, without restricting the number of bits they are allowed to read.

**Claim 4.1** *Let $A$ be a recovery algorithm for any code $\mathbf{C}$: $\{0,1\}^n \to \{0,1\}^m$. If $\delta \ge \frac{1}{2}$, then $\zeta^\star_\delta(A) \le \frac{1}{2}$.*

**Proof:** Arbitrarily choose $i \in [n]$. Take any $a, b \in \{0,1\}^n$ such that $a_i \neq b_i$. Construct $Y \in \{0,1\}^m$ as being the same as $\mathbf{C}(a)$ on the first $\frac{m}{2}$ positions and the same as $\mathbf{C}(b)$ on the last $\frac{m}{2}$ positions. Whenever the input to $\mathbf{C}$ is $a$ or $b$, the adversary will corrupt the codeword to $Y$ by modifying $\frac{m}{2}$ positions. Either $\Pr[A^Y(i) \text{ outputs } 1] \leq \frac{1}{2}$ or $\Pr[A^Y(i) \text{ outputs } 1] \geq \frac{1}{2}$, where the probabilities are over the internal coin tosses of $A$. In the first case, the algorithm fails with probability at least $\frac{1}{2}$ on whichever input $a$ or $b$ has $i$'th position 1. In the second case, the algorithm fails with probability at least $\frac{1}{2}$ on whichever input $a$ or $b$ has $i$'th position 0. Thus, in either case, we have shown there exists an input and an adversary error pattern of size at most $\delta m$ so that the probability of error is at least $\frac{1}{2}$. So $\zeta_\delta(A) \leq \frac{1}{2}$. ■

Now we will generalize to any input alphabet, not just $\{0,1\}$.

**Claim 4.2** *Let $A$ be a recovery algorithm for any code $\mathbf{C}$: $\Sigma^n \to \Sigma^m$. If $\delta \geq 1 - \frac{1}{|\Sigma|}$, then $\zeta^\star_\delta(A) \leq \frac{1}{|\Sigma|}$.*

**Proof:** Arbitrarily choose $i \in [n]$. For $s \in \Sigma$, let $g_s \in \Sigma^n$ be vectors such that $(g_s)_i = s$. Split $[m]$ into a partition of $|\Sigma|$ equal sized subsets named $U_s$ for $s \in \Sigma$. Construct $Y \in \Sigma^m$ in the following way. For each $s \in \Sigma$, let $Y$ be the same as $\mathbf{C}(g_s)$ on the positions in $U_s$. Whenever the input to $\mathbf{C}$ is one of the $g$'s, the adversary will corrupt the codeword to $Y$ by modifying at most $m(1 - \frac{1}{|\Sigma|})$ positions. Now

$$\sum_{s \in \Sigma} \Pr[A^Y(i) \text{ outputs } s] = 1$$

where the probability is over the internal coin tosses of $A$. So there exists at least one $s \in \Sigma$ such that, if the adversary corrupts $\mathbf{C}(g_s)$ into $Y$, the probability of algorithm correctly answering $s$ is at most $\frac{1}{|\Sigma|}$. Therefore, we have shown there exists an input $x$ and an adversary error pattern of size at most $\delta m$ so that the probability of error is at least $1 - \frac{1}{|\Sigma|}$. So $\zeta_\delta(A) \leq \frac{1}{|\Sigma|}$. ■

Here are some other observations that hold regardless of the number of queries an algorithm queries.

**Lemma 4.3** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a* $(q, \delta, \epsilon)$*-LDC with* $\epsilon > 0$*. Then* $\mathbf{C}$ *has minimum distance at least* $2\delta m + 1$*.*

**Proof:** We prove the contrapositive. Assume there are two codewords $\mathbf{C}(a)$ and $\mathbf{C}(b)$ with $a \neq b \in \{0,1\}^n$ such that the Hamming distance between them is less than $2\delta m + 1$. Because $a \neq b$, $a$ and $b$ differ in at least one bit – without loss of generality, let $i \in [n]$ be one such bit in the support of $a - b$. Because $d(\mathbf{C}(a), \mathbf{C}(b)) \leq 2\delta m$, there exists a string, call it $Y$, such that $d(\mathbf{C}(a), Y) \leq \delta m$ and $d(Y, \mathbf{C}(b)) \leq \delta m$. Whenever the input to the code is $a$ or $b$, the adversary will change the codeword into $Y$. Either $\Pr[A^Y(i) \text{ outputs } 1] \leq \frac{1}{2}$ or $\Pr[A^Y(i) \text{ outputs } 1] \geq \frac{1}{2}$, where the probabilities are over the internal coin tosses of $A$. In the first case, the algorithm fails with probability at least $\frac{1}{2}$ on whichever input $a$ or $b$ has $i$'th position 1. In the second case, the algorithm fails with probability at least $\frac{1}{2}$ on whichever input $a$ or $b$ has $i$'th position 0.

Thus, in either case, we have shown there exists an input and an adversary error pattern of size at most $\delta m$ so that the probability of error is at least $\frac{1}{2}$, which contradicts the assumption that $\epsilon > 0$. ■

If we restrict to linear codes, we can generalize Lemma 4.3 to any field. Note that when we work with codes over a field $F$, addition and multiplication operations are performed over $F$.

**Lemma 4.4** *Let* $\mathbf{C}$*:* $F^n \to F^m$ *be a linear* $(q, \delta, \epsilon)$*-LDC with* $\epsilon > 0$*. Then* $\mathbf{C}$ *has minimum distance at least* $\frac{|F|}{|F|-1}\delta m + 1$*.*

**Proof:** We prove the contrapositive. Assume there are two codewords $\mathbf{C}(g_0)$ and $\mathbf{C}(g_1)$ with $g_0 \neq g_1 \in F^n$ such that the Hamming distance between them is less than $\frac{|F|}{|F|-1}\delta m + 1$. Define the following:

$$\forall f \in F, f \neq 0, 1 : g_f \triangleq g_0 + f(g_1 - g_0)$$

(Clearly, the notation $g_0$ and $g_1$ for the first two codewords was chosen for consistency with the definitions of the other $g_f$.)

Because $g_0 \neq g_1$, $g_0$ and $g_1$ differ in at least one bit – without loss of generality, let $i \in [n]$ be one such bit in the support of $g_0 - g_1$. For $f \in F$, define $h_f$ as the unique $g_{f'}$ ($f' \in F$) such $(g_{f'})_i = f$.

Construct a string $Y$ in the following way. In the positions outside the support of $\mathbf{C}(g_0) - \mathbf{C}(g_1)$, let $Y$ equal $\mathbf{C}(g_0)$. (Notice for later that because $\mathbf{C}$ is linear, the $\mathbf{C}(g_f)$ are identical outside of the support of $\mathbf{C}(g_0) - \mathbf{C}(g_1)$.) Divide the positions in the support of $\mathbf{C}(g_0) - \mathbf{C}(g_1)$ into $|F|$ equal pieces and label each piece by a member of $F$. For the positions in the $f \in F$ piece, let $Y$ be the same as $h_f$. This implies $\forall f \in F, d(\mathbf{C}(h_f), Y) \leq \frac{|F|-1}{|F|}\frac{|F|}{|F|-1}\delta m = \delta m$. Whenever the input to the code is $h_f$, for some $f \in F$, the adversary will change the codeword into $Y$. Now

$$\sum_{f \in F} \Pr[A^Y(i) \text{ outputs } f] = 1$$

where the probability is over the internal coin tosses of $A$. So there exists at least one $f \in F$ such that, if the adversary corrupts $\mathbf{C}(h_f)$ into $Y$, the probability of algorithm correctly answering $f$ is at most $\frac{1}{|F|}$. Therefore, we have shown there exists an input $x$ and an adversary error pattern of size at most $\delta m$ so that the probability of error is at least $1 - \frac{1}{|F|}$, which contradicts the assumption that $\epsilon > 0$. ■

These results about the minimum distance of an LDC can be used with common bounds from coding theory. The Plotkin bound implies that a code with alphabet $\Sigma$ and minimum distance larger than $\frac{|\Sigma|-1}{|\Sigma|} + \alpha$ for any constant $\alpha > 0$ has at most a constant number of codewords. Therefore, Lemma 4.3 implies that a binary $(q, \delta, \epsilon)$-LDC with $\epsilon > 0$ must have $\delta \leq \frac{1}{4} + O(1/n)$. Similarly, Lemma 4.4 implies that a linear $(q, \delta, \epsilon)$-LDC over field $F$ with $\epsilon > 0$

must have $\delta \leq (\frac{|F|-1}{|F|})^2 + O(1/n)$.

We can apply other common bounds from coding theory as well. For binary codes with $\epsilon > 0$, Lemma 4.3 and the Hamming bound give:

**Lemma 4.5** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a* $(q, \delta, \epsilon)$*-LDC with arbitrary* $q$ *and* $\epsilon > 0$*. Then* $n \leq m - \log \sum_{i=0}^{\delta m} \binom{m}{i}$.

**Proof:** Let $d$ be the minimum distance of the code in question, where, because of Lemma 4.3, $d \geq 2\delta m + 1$. Then,

$$n \leq m - \log \sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{m}{i}$$

$$\Rightarrow n \leq m - \log \sum_{i=0}^{\delta m} \binom{m}{i}$$

■

This implies an upper limit for $\delta$. Alternatively, this also implies a lower bound on $m$ in terms of $\delta$ and $n$. There is an analogous Hamming bound for non-binary codes. If we assume linearity, we can use it together with Lemma 4.4 to get similar results.

Now let us consider another well known coding theory bound, the Singleton bound. For binary codes having $\epsilon > 0$, Lemma 4.3 and the Singleton bound give:

**Lemma 4.6** *Let* $\mathbf{C}$*:* $\{0,1\}^n \rightarrow \{0,1\}^m$ *be a* $(q, \delta, \epsilon)$*-LDC with arbitrary* $q$ *and* $\epsilon > 0$*. Then*

$$\delta \leq \frac{1}{2}(1 - \frac{n}{m})$$

**Proof:** The Singleton bound states that the minimum distance of a code is less than or equal to the length of that code minus the number of information bits in that code plus one. We have lower bounded the minimum distance of a binary code having $\epsilon > 0$ in Lemma 4.3. Thus,

$$2\delta m + 1 \leq m - n + 1$$
$$\Rightarrow \delta \leq \frac{1}{2}(1 - \frac{n}{m})$$

■

For linear codes over field $F$ having $\epsilon > 0$, Lemma 4.4 and the Singleton bound give:

**Lemma 4.7** *Let* $\mathbf{C}$*:* $F^n \rightarrow F^m$ *be a linear,* $(q, \delta, \epsilon)$*-LDC with arbitrary* $q$ *and* $\epsilon > 0$*. Then*

$$\delta \leq \frac{|F| - 1}{|F|}(1 - \frac{n}{m})$$

**Proof:** The Singleton bound states that the minimum distance of a code is less than or equal to the length of that code minus the number of information bits in that code plus one. We have lower bounded the minimum distance of

a linear code having $\epsilon > 0$ in Lemma 4.4. Thus,

$$\frac{|F|}{|F|-1}\delta m + 1 \leq m - n + 1$$
$$\Rightarrow \delta \leq \frac{|F|-1}{|F|}(1 - \frac{n}{m})$$

■

Here is a generalization of Theorem 2 from Katz and Trevisan [32] for functions over arbitrary alphabets. After proving it, we will see how it yields a new impossibility result. We also use this theorem in the proof of Claim 5.3.3.

For the proof of the theorem, we will use the following result, which was originally proved by Robert Fano. Our version is adapted from Cover and Thomas [14]):

**Theorem 4.8** *(Robert Fano.) Let $X$ be a random variable in the domain $\tilde{X}$. For any estimator $\hat{X}$ of $X$ such that $X \to Y \to \hat{X}$ with $P_e \triangleq \Pr(X \neq \hat{X})$, we have*

$$H(P_e) + P_e \log |\tilde{X}| \geq H(X \mid Y)$$

Note that, as is usual terminology, $H$ represents both the entropy function on a random variable and the binary entropy function (the meaning being clear from context).

**Theorem 4.9** *Let* $\mathbf{C}$*:* $\Sigma^n \to R$ *be a function. Assume there is an algorithm* $A$ *such that:*

$$\forall i \in [n] : \Pr_x[A(\mathbf{C}(x), i) = x_i] \geq \frac{1}{|\Sigma|} + \epsilon$$

*Then* $\log |R| \geq n\Big(\log |\Sigma| - H(\frac{1}{|\Sigma|} + \epsilon) - (\frac{1}{|\Sigma|} + \epsilon)\log(|\Sigma| - 1)\Big)$.

**Proof:** Let $x$ be chosen uniformly at random from $\Sigma^n$. Consider the mutual information between $x$ and $\mathbf{C}(x)$:

$$I(x; \mathbf{C}(x)) \leq H(\mathbf{C}(x)) \leq \log |R|$$

On the other hand,

$$\begin{aligned}
&I(x; \mathbf{C}(x)) = H(x) - H(x \mid \mathbf{C}(x)) \\
&\geq H(x) - \sum_{i \in [n]} H(x_i \mid \mathbf{C}(x)) \\
&\geq n\Big(\log |\Sigma| - H(\frac{1}{|\Sigma|} + \epsilon) - (\frac{1}{|\Sigma|} + \epsilon)\log(|\Sigma| - 1)\Big) \quad \text{Fano's Inequality}
\end{aligned}$$

■

Theorem 3 from [32] states that one query LDCs with binary inputs and small enough output alphabet sizes do not exist when $n$ is large enough. This can easily be generalized by using Theorem 4.9 here instead of the Theorem 2 in their paper:

**Theorem 4.10** *Let* $\mathbf{C}$*:* $\Sigma^n \to \Gamma^m$ *be a* $(1, \delta, \epsilon)$*-LDC. Then:*

$$n \leq \frac{\log |\Gamma|}{\delta\Big(\log |\Sigma| - H(\frac{1}{|\Sigma|} + \epsilon) - (\frac{1}{|\Sigma|} + \epsilon)\log(|\Sigma| - 1)\Big)}$$

This shows that a one query LDC, no matter what the input alphabet is, does not exist when the output alphabet size is small enough and $n$ is large enough. This is a new impossibility result.

# Chapter 5

# Technical Tools

This chapter contains technical tools that are common to many of our proofs.

## 5.1 Properties of Query Sets for Linear Codes

In this section, we show that the query sets for linear LDCs that achieve high enough correctness have certain interesting properties.

The following theorem is very important, because it shows that, if the algorithm hopes to do better than random guessing, it must query sets that have a special mathematical property. It was stated just for the two query case in Lemma 3.2 of Goldreich et al. [24].

**Theorem 5.1.1** *Consider a linear code* $\mathbf{C}$ *and a* $q$ *query algorithm* $A$*. Let* $i \in [n]$ *and* $Q = \{j_1, j_2, ...j_{q'}\} \subset [m]$ *for* $q' \leq q$ *such that* $A$ *queries* $Q$ *with nonzero probability when tasked to find input position* $i$*. Suppose*

$$\Pr_{x \epsilon_U F^n}[A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ queries } Q] > \frac{1}{|F|}$$

*where the probability is taken over letting $x$ be uniformly random from $F^n$ and over the internal coin tosses of $A$. Then there exist $c_{j_1}, c_{j_2}, ...c_{j_{q'}} \in F$ such that $\sum_{k=1}^{q'} c_{j_k} a_{j_k} = e_i$.*

**Proof:** We prove the contrapositive. Take any $i$ and $Q = \{j_1, j_2, ...j_{q'}\} \subset [m]$ such that there do not exist $c_{j_1}, c_{j_2}, ...c_{j_{q'}} \in F$ for which $\sum_{k=1}^{q'} c_{j_k} a_{j_k} = e_i$. Say $y_{j_1}$, $y_{j_2}$, ..., and $y_{j_{q'}}$ are the respective values $A$ receives from querying $Q$. The algorithm's job when it queries $Q$ is to solve the following system of $q'$ linear equations for $x_i$:

$$y_{j_1} = a_{j_1} \cdot x$$

$$y_{j_2} = a_{j_2} \cdot x$$

$$\vdots$$

$$y_{j_{q'}} = a_{j_{q'}} \cdot x$$

Assume without loss of generality that $a_{j_1}$, $a_{j_2}$, ... $a_{j_{q''}}$ is a maximal collection of linearly independent vectors from $a_{j_1}$, $a_{j_2}$, ... $a_{j_{q'}}$, for some $q'' \leq q'$. (Simply renumber the $a$'s so this is true.) Therefore, the system of $q'$ linear equations above turns into a system of $q''$ independent linear equations.

Because the vector $e_i$ is not in the span of $\{a_{j_1}, a_{j_2}, ...a_{j_{q''}}\}$, there exists

an $\hat{x}$ satisfying:

$$e_i \cdot \hat{x} = 1$$

$$a_{j_1} \cdot \hat{x} = 0$$

$$a_{j_2} \cdot \hat{x} = 0$$

$$\vdots$$

$$a_{j_{q''}} \cdot \hat{x} = 0$$

(Because this is a system of $q'' + 1$ independent linear equations, a solution $\hat{x}$ must exist.) Note that $\hat{x} \neq 0$ because $\hat{x}_i \neq 0$. Using $\hat{x}$, we define the following set:

$$V_x \triangleq \{x, x + \hat{x}, x + 2\hat{x}, ..., x + (|F| - 1)\hat{x}\}$$

For any $x$ that is a solution of the original $q'$ equations, every member of $V_x$ is a solution as well, but each has a different $i$'th coordinate. Notice for any $x' \notin V_x$, $V_x \cap V_{x'} = \varnothing$. This implies that the number of solutions to the original $q'$ equations having $i$'th coordinate equal $d$, for any $d \in F$, is the same. Recall that we are considering uniform $x \in F^n$. So,

$$\forall d \in F : \Pr_{x \epsilon_U F^n}[x_i = d \mid y_{j_1} = a_{j_1} \cdot x, y_{j_2} = a_{j_2} \cdot x, \text{ ..., and } y_{j_{q'}} = a_{j_{q'}} \cdot x] = \frac{1}{|F|}$$

This implies

$$\Pr_{x \epsilon_U F^n}[A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ queries } Q] = \frac{1}{|F|}$$

This contradicts the assumption from the theorem's statement that the probability of correctness is strictly greater than $\frac{1}{|F|}$. ■

We will need the following simple fact in many of our correctness and lower bound proofs.

**Fact 5.1.2** *Let $a_1$, $a_2$, ... $a_t$ be vectors from $\{0,1\}^n$. Let $X$ be uniformly random over $\{0,1\}^n$. Then $a_1 \cdot X$, $a_2 \cdot X$, ... $a_t \cdot X$ are $t$ independent, uniformly random bits if and only if $a_1$, $a_2$, ... $a_t$ are linearly independent over $F_2$.*

**Proof:** Let us consider what happens when $a_1$, $a_2$, ... $a_t$ are linearly independent over $F_2$. Then, for any set of values $d_1, d_2, ...d_t \in \{0,1\}$, the number of $x$ such that $\forall 1 \leq i \leq t$, $a_i \cdot x = d_i$ is always the same: $2^{n-t}$. Since the probability that $X = x$ is the same for any $x$, $a_1 \cdot X$, $a_2 \cdot X$, ... $a_t \cdot X$ are $t$ independent, uniformly random bits.

If $a_1$, $a_2$, ... $a_t$ are not linearly independent, then there exist $c_1, c_2, ...c_t \in \{0,1\}$ (not all zero) such that $\sum_{k=1}^{t} c_k a_k = 0$ (where the sum is over $F_2$). This means that any set of values for $a_1 \cdot X$, $a_2 \cdot X$, ... $a_t \cdot X$ for which $\sum_{k=1}^{t} c_k a_k \cdot X = 1$ will never appear. In particular, if $t'$ is such that $c_{t'} \neq 0$, then the set of values $\forall i \neq t'$, $a_i \cdot X = 0$ and $a_{t'} \cdot X = 1$ never appears. Thus, $a_1 \cdot X$, $a_2 \cdot X$, ... $a_t \cdot X$ cannot be $t$ independent, uniformly random bits. ■

A small extension of this fact we will use frequently is the following. If $X$ is uniformly random over $\{0,1\}^n$, $a_1$, $a_2$, ... $a_t$ are linearly independent

over $F_2$, and $b_1$, $b_2$, ... $b_t$ are fixed members of $F_2$, then $a_1 \cdot X + b_1$, $a_2 \cdot X + b_2$, ... $a_t \cdot X + b_t$ are also $t$ independent, uniformly random bits.

## 5.2 Probabilistic Adversary

Locally decodable codes have a very strong error model: they must answer any requested input bit with a high probability when the codeword has been corrupted by an arbitrary error of a limited size. Because of this, strong limitations of LDCs, such as for correctness and codeword length, can be found by considering certain corruption patterns. Most papers studying LDCs have not taken advantage of this, however, because they immediately reduce LDCs to smooth codes, which have no corruption model. Two papers that do not make this reduction and study the error patterns of LDCs are Obata [38] and Shiowattana and Lokam [42]. Both papers are for LDCs whose algorithms query at most two positions. These two papers are based on considering strategies for the adversary to corrupt positions of the codeword.

This section provides several techniques that advance on previous methods and allow us to create strong adversaries. These adversaries have importance in defeating algorithms querying more than two positions. To start, here is an important technical lemma that allows us to simplify many proofs.

**Lemma 5.2.1** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a code. Choose two disjoint subsets* $R, S \subseteq [m]$*. Let* $B_1$ *be chosen from* $R$ *under an arbitrary probability*

*distribution, and let $B_2$ be chosen uniformly at random from all subsets of $S$. For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Fix $i \in [n]$, a natural number $q$, and a positive real $\epsilon$. Assume there exists a $q$ query LDC recovery algorithm $A$ achieving*

$$\Pr_{x \in F^n, B}[A^{\mathbf{C}(x)+B}(i) = x_i] \geq \frac{1}{2} + \epsilon$$

*where the probability is over the internal coin tosses of $A$, uniform $x$, and $B$ chosen by the product distribution of the distributions of $B_1$ and $B_2$. Then there exists a $q$ query LDC recovery algorithm $\tilde{A}$ achieving*

$$\Pr_{x \in F^n, B}[\tilde{A}^{\mathbf{C}(x)+B}(i) = x_i] \geq \frac{1}{2} + \epsilon$$

*as well, and $\tilde{A}$ never queries any positions from $S$.*

**Proof:** Note that $R$ is irrelevant to what is below.

Without loss of generality, say that $S = [t]$ for some $0 \leq t \leq m$. For any $x \in F^n$ and $s'_1$, $s'_2$, ... $s'_t \in \{0, 1\}$, there exists exactly one sequence of $s_1$, $s_2$, ... $s_t$ such that

$$\mathbf{C}(x)_1 + s_1 = s'_1$$

$$\mathbf{C}(x)_2 + s_2 = s'_2$$

$$\vdots$$

$$\mathbf{C}(x)_t + s_t = s'_t$$

Each sequence $s_1$, $s_2$, ... $s_t$ has the same probability of occurring as the characteristic vector of $B_2$ ($\frac{1}{2^t}$). So any values the algorithm receives from the positions labeled by members of $S$ are independent, uniformly random values from $\{0,1\}$. Therefore, we can construct a new algorithm $\tilde{A}$ that behaves exactly as $A$ except, whenever $A$ queries a member of $S$, $\tilde{A}$ samples uniformly at random from $\{0,1\}$ instead. Then over random $x$ and $B$ as above, when the coin tosses of $A$ and $\tilde{A}$ are fixed and equal, the distribution of values $A$ and $\tilde{A}$ receive as answers to their queries are the same. Thus, $\tilde{A}$ achieves the same correctness as $A$ under random $x$ and $B$. ■

Now we consider two basic probability distributions. Then we will see how successful recovery algorithms can be if the adversary is allowed to use these distributions to corrupt codes.

**Definition 5.2.1** *Take a ground set* $[m]$ *for* $m \geq 1$. *The probability distribution of the random variable* $B \subseteq [m]$ *is said to be BINOMIAL with probability* $\delta$ *if each member of* $[m]$ *is chosen to be part of* $B$ *independently with probability* $\delta$.

**Definition 5.2.2** *Take a ground set* $[m]$ *for* $m \geq 1$ *and an* $s \leq m$. *The probability distribution of the random variable* $B \subseteq [m]$ *is said to be FIXED SIZE with size* $s$ *if* $B$ *is chosen uniformly from all subsets of size* $s$ *from* $[m]$.

These are standard, well studied distributions. We intend to use these distributions as building blocks to create often times more complex distributions that will decide which positions of an LDC codeword the adversary corrupts. The first point to be made is that, when analyzing the effect of an adversary's actions on the performance of an LDC algorithm, it is usually more mathematically tractable to assume the adversary uses a derivative of the first distribution than a derivative of the second. However, in the LDC model, the adversary is limited to corrupting no more than $\delta$ fraction of the codeword. This can be a problem for derivatives of the first distribution, because the number of positions chosen in the binomial distribution is not fixed. Random graph theory, for instance, grapples with the same decision in the context of choosing edges among pairs of vertices. Results there show that as $m$ gets large, the binomial and the fixed $\delta m$ size distributions are asymptotically equivalent in many situations. We can take advantage of a similar asymptotic equivalence.

Let us be more precise. With LDCs, we consider the probability that $q$ sized subsets (edges) of $[m]$ are corrupted. To that end, consider the probability that an adversary using the binomial distribution with probability $\delta$ chooses exactly $k$ positions of a specific $q$ sized subset of $[m]$:

$$\binom{q}{k}\delta^k(1-\delta)^{q-k}$$

Now consider the probability that an adversary using the fixed $\delta m$ size

distribution chooses exactly $k$ positions of a specific $q$ sized subset of $[m]$:

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}}$$

We provide upper and lower bounds on this second expression in terms of the first the following two lemmas. The lower bound in particular will be an essential tool in proving some of our bounds on correctness and lower bounds on size.

**Lemma 5.2.2** *For large enough $m$,*

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} > \binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$$

**Proof:** When $1 \leq k < q$:

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} = \binom{q}{k}\frac{(\delta m)!(m-\delta m)!}{m!}\frac{(m-q)!}{(\delta m-k)!(m-\delta m-q+k)!} = \binom{q}{k}\cdot$$
$$\frac{\delta m(\delta m-1)...(\delta m-k+1)(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)}{m(m-1)...(m-q+1)}$$

Let us start by considering just the numerator:

$$\delta m(\delta m-1)...(\delta m-k+1)(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)$$

Expand it out, and collect like powers of $m$. This gives:

$$\delta^k(1-\delta)^{q-k}m^q - \Big((\sum_{i=0}^{k-1} i)\delta^{k-1}(1-\delta)^{q-k} + (\sum_{i=0}^{q-k-1} i)\delta^k(1-\delta)^{q-k-1}\Big)m^{q-1} + O(m^{q-2}) \tag{5.1}$$

Note that in the sum above, the $m^{q-j}$ terms have positive coefficients when $j$ is even and negative coefficients when $j$ is odd. Therefore, for large enough $m$, we can bound the numerator by just two terms:

$$> \delta^k(1-\delta)^{q-k}m^q - \Big((\sum_{i=0}^{k-1} i)\delta^{k-1}(1-\delta)^{q-k} + (\sum_{i=0}^{q-k-1} i)\delta^k(1-\delta)^{q-k-1}\Big)m^{q-1}$$

$$= \delta^k(1-\delta)^{q-k}m^q - \Big(\frac{(k-1)k}{2}\delta^{k-1}(1-\delta)^{q-k} + \frac{(q-k-1)(q-k)}{2}\delta^k(1-\delta)^{q-k-1}\Big)m^{q-1}$$

replacing the summations

$$> \delta^k(1-\delta)^{q-k}m^q - \frac{q^2}{2}(\delta+1-\delta)\delta^{k-1}(1-\delta)^{q-k-1}m^{q-1}$$

since $k(k-1)$ and $(q-k-1)(q-k)$ are both less than $q^2$

$$> \delta^k(1-\delta)^{q-k}m^q - q^2m^{q-1}$$

The denominator, $m(m-1)...(m-q+1)$, is upper bounded by $m^q$, so the overall expression is bounded by

$$> \binom{q}{k}\frac{\delta^k(1-\delta)^{q-k}m^q - q^2m^{q-1}}{m^q}$$

$$= \binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$$

The analysis from the $1 \leq k < q$ case applies to the $k = 0$ case as well, except that the $(\sum_{i=0}^{k-1} i)\delta^{k-1}(1-\delta)^{q-k}$ term in (5.1) is not present. The coefficient in front of $m^{q-1}$ can be lower bounded in the same way. So $\binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$ is a lower bound for the $k = 0$ case as well.

When $k = q$:

$$\begin{aligned}
\frac{\binom{m-q}{\delta m-q}}{\binom{m}{\delta m}} &= \frac{\delta m(\delta m-1)...(\delta m-q+1)}{m(m-1)...(m-q+1)} \\
&= \frac{\delta^q m^q - \frac{(q-1)q}{2}\delta^{q-1}m^{q-1} + O(m^{q-2})}{m(m-1)...(m-q+1)} \\
&> \frac{\delta^q m^q - q^2 m^{q-1}}{m(m-1)...(m-q+1)} \\
&> \frac{\delta^q m^q - q^2 m^{q-1}}{m^q} \\
&= \delta^q - \frac{q^2}{m}
\end{aligned}$$

■

**Lemma 5.2.3** *For large enough $m$,*

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} < \binom{q}{k}\delta^k(1-\delta)^{q-k} + \frac{2q^2}{m}$$

**Proof:** When $0 \leq k < q$:

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} = \binom{q}{k}\frac{(\delta m)!(m-\delta m)!}{m!}\frac{(m-q)!}{(\delta m-k)!(m-\delta m-q+k)!} = \binom{q}{k}\cdot$$

$$\frac{\delta m(\delta m-1)...(\delta m-k+1)(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)}{m(m-1)...(m-q+1)}$$

$$< \binom{q}{k}\frac{(\delta m)^k(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)}{m(m-1)...(m-q+1)}$$

$$\leq \binom{q}{k}\frac{(\delta m)^k(m-\delta m)^{q-k}}{m^{q-k}(m-q+k)(m-q+k-1)...(m-q+1)}$$

$$< \binom{q}{k}\frac{(\delta m)^k(m-\delta m)^{q-k}}{m^{q-k}(m-q)^k}$$

$$= \binom{q}{k}\delta^k(1-\delta)^{q-k}\frac{1}{(1-\frac{q}{m})^k}$$

$$< \binom{q}{k}\delta^k(1-\delta)^{q-k}\frac{1}{1-\frac{kq}{m}} \qquad \text{for } m \text{ large enough}$$

$$< \binom{q}{k}\delta^k(1-\delta)^{q-k}(1+\frac{2kq}{m}) \qquad \text{for } m \text{ large enough}$$

$$\leq \binom{q}{k}\delta^k(1-\delta)^{q-k}+\frac{2q^2}{m} \qquad \text{because } \binom{q}{k}\delta^k(1-\delta)^{q-k} \leq 1$$

When $k = q$:

$$\frac{\binom{m-q}{\delta m-q}}{\binom{m}{\delta m}} = \frac{\delta m(\delta m-1)...(\delta m-q+1)}{m(m-1)...(m-q+1)} < (\frac{\delta m}{m})^q = \delta^q$$

■

Here is a technical lemma that will be important in several of our proofs. It holds for any distribution of the adversary as long as it is independent of the input.

**Definition 5.2.3** *We use the notation* $(\mathbf{C}(x)+B)_Q$ *to mean the values of the codeword* $\mathbf{C}(x)$ *corrupted by* $B$ *on the positions indexed by the query set* $Q$*.*

A crucial, but subtle, point used in the proof below is that, for any $Q$, the value of $(\mathbf{C}(x) + B)_Q$ is independent of the event $A$ queries $Q$.

**Lemma 5.2.4** *Let* $\mathbf{C}$ *be a code* $\Sigma^n \to \Sigma^m$ *and* $A$ *be an LDC recovery algorithm for* $\mathbf{C}$. *Let* $x$ *be a uniformly random value from* $\Sigma^n$ *and* $B$ *be a random corruption of the codeword* $\mathbf{C}(x)$. *Assume the distributions of* $x$ *and* $B$ *are independent. Finally, let* $E$ *be an event that does not depend on the internal randomness of* $A$. *Then, for any* $i \in [n]$, $Q \subset [m]$, $v \in \Sigma$, *and any bit string* $s$ *(representing the answers to the queries* $A$ *makes):*

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s] = \Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = s]$$

**Proof:** Without loss of generality, assume $A$ flips all of its coin flips in advance of querying any codeword positions. Let $r$ denote the event that the outcome of these coin flips is a particular string $r$. Then we can break apart the first probability in the equation above:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s] = \sum_r \Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s; r] \cdot \Pr_{x,B}[r \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s]$$

Now $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s; r]$ is either 0 or 1. Therefore, we can remove the conditioning on $E$ from $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) =$

$v \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s; r]$ and get:

$$\begin{aligned}
&\Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s] \\
&\qquad = \sum_r \Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s; r] \cdot \\
&\qquad\qquad \Pr_{x,B}[r \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s]
\end{aligned}$$

Let us analyze just the second term of the last expression. For any $r$,

$$\begin{aligned}
&\Pr_{x,B}[r \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s] \\
&\qquad = \frac{\Pr_{x,B}[r; A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s]}{\Pr_{x,B}[A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s]} \\
&\qquad = \frac{\Pr_{x,B}[r; A \text{ queries } Q] \Pr_{x,B}[E; (\mathbf{C}(x)+B)_Q = s]}{\Pr_{x,B}[A \text{ queries } Q] \Pr_{x,B}[E; (\mathbf{C}(x)+B)_Q = s]} \\
&\qquad = \frac{\Pr_{x,B}[r; A \text{ queries } Q]}{\Pr_{x,B}[A \text{ queries } Q]}
\end{aligned}$$

Above, we have used the fact that $r$ and $A$ queries $Q$ is independent of $E$ and the values of $\mathbf{C}(x)+B$ on the positions indexed by $Q$. By similar logic, $\Pr_{x,B}[r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] = \frac{\Pr_{x,B}[r; A \text{ queries } Q]}{\Pr_{x,B}[A \text{ queries } Q]}$. Therefore,

$$\begin{aligned}
&\Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q = s] \\
&\qquad = \sum_r \Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s; r] \cdot \\
&\qquad\qquad \Pr_{x,B}[r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] \\
&\qquad = \Pr_{x,B}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s]
\end{aligned}$$

■

## 5.3 Linear Decoders

We will wish to discuss what linearity means to an LDC. Usually, when a code is called linear, it means that its positions are each linear combinations of the input positions. However, in the LDC literature, that concept of linearity is sometimes confused with the concept of the possible linearity of the decoding algorithm. To explain more, we present the following definition. It is natural given earlier work, but we make it explicit:

**Definition 5.3.1** *Let* $\mathbf{C}$ *be an arbitrary (possibly non-linear) code. We say that an algorithm A is a LINEAR DECODER for* $\mathbf{C}$ *if, for any fixing of the outcomes of the coin flips of A, the value it returns is a fixed linear combination of the codeword positions it reads.*

Note that allowed in the definition of linear decoders is the strategy of ignoring the codeword positions read and just returning the result of a coin flip. Also note that a non-linear code can have a linear decoder. We will only consider linear decoders that are non-adaptive.

Many algorithms presented in the literature are linear decoders, so we prove several results about then. Our first two results will be about linear decoders that are required to query a certain number of positions. They can be considered warm-ups to the more general results that follow later.

We can use Lemma 5.2.2 to quickly prove the following two correctness bounds.

**Claim 5.3.1** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a binary linear code with* $n$ *large enough. For any linear decoder* $A$ *using exactly* $q$ *positions,* $\zeta^\star_\delta(A) \leq 1 - \sum_{k=1,k\ odd}^{q} \binom{q}{k}\delta^k(1-\delta)^{q-k} + \frac{q^3}{n}$.

**Proof:** The adversary chooses a set of $\delta m > q$ positions uniformly at random from $[m]$. The adversary adds 1 to each of these positions. Now let us analyze the probability of error of the algorithm. Error happens when the edge the algorithm selects intersects with the adversary's chosen positions in $k$ positions, for odd $k$. For each $k$, the probability of error is the same for each edge the algorithm could query. A lower bound for that probability is given by Lemma 5.2.2. Summing that result for odd $k$, and using the fact that the lower bound of Katz and Trevisan [32] implies, for large enough $n$, $m > n$, the result is obtained. ■

**Claim 5.3.2** *Let* $\mathbf{C}$*:* $F^n \to F^m$ *be a code with* $n$ *large enough. For any linear decoder* $A$ *using exactly* $q \geq 1$ *positions,*

$$\zeta^\star_\delta(A) \leq 1 - \sum_{k=1}^{q}\Big(\Big(\binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^3}{n}\Big)\sum_{i=1}^{k}\Big(\frac{-1}{|F|-1}\Big)^{i-1}\Big)$$

**Proof:** The adversary chooses a set of $\delta m$ positions uniformly at random from $[m]$. The adversary corrupts each of these positions to a random value from $F$ different from what it was uncorrupted. Each random corruption is

chosen independently. Now let us analyze the probability of error of the algorithm. The probability of error is the same for each edge, so choose some particular edge.

For a given $1 \leq k \leq q$, the probability that exactly $k$ of the vertices in that edge are corrupted is, by Claim 5.2.2, at least $\binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$. The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. So the probability is at least $\binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{n}$.

We step back to prove a simple induction. Conditioned on the event that exactly one vertex in the edge is corrupted, the probability that the decoding formula returns the wrong answer is 1. Conditioned on the event that exactly $k$ (for $2 \leq k \leq q$) vertices of the edge are corrupted, the probability that the decoding formula returns the wrong answer is $z(1 - \frac{1}{|F|-1}) + (1-z)$, where $z$ is the probability that the decoding formula returns the wrong answer conditioned on the event that exactly $k-1$ vertices of the edge are corrupted. This implies that, conditioned on the event that exactly $k$ (for $1 \leq k \leq q$) vertices of the edge are corrupted, the probability that the decoding formula returns the wrong answer is $\sum_{i=1}^{k}(\frac{-1}{|F|-1})^{i-1}$.

Multiplying the probabilities found in paragraphs two and three and summing over $1 \leq k \leq q$ gives the answer. ■

The bound proven in this claim can be approximated as $1 - q\delta + o(\delta)$.

Now let us prove a related result for more general linear decoders. This statement is very general in that it works for any number of queries and any field size.

**Claim 5.3.3** *Let* $\mathbf{C}$*:* $\Sigma^n \to \Sigma^m$ *be a code with* $n$ *large enough. For any linear decoder* $A$*,*

$$\zeta_\delta(A) \leq \max\Big(1 - 2\delta + \frac{|\Sigma|}{|\Sigma| - 1}\delta^2 + \frac{9}{\sqrt{n}} + \frac{24}{n}, \frac{1}{|\Sigma|} + \frac{1}{n^{.25}}\Big)$$

**Proof:** For simplicity, define $t \triangleq \frac{1}{n^{.25}}$ in the following.

For $i \in [n]$, define:

$$R_i \triangleq \Big\{ j \in [m] \;\Big|\; \Pr_x[A^{\mathbf{C}(x)}(i) \text{ is correct} \mid A \text{ queries } \{j\}] > \frac{1}{|\Sigma|} + t \Big\}$$

where the probability is taken with respect to uniform $x$ in $\Sigma^n$. Now consider:

$$S \triangleq \Big\{ i \in [n] \;\Big|\; |R_i| \geq \nu m \Big\}$$
$$\text{for } \nu \triangleq \frac{\log|\Sigma|}{.99n(\log|\Sigma| - H(\frac{1}{|\Sigma|} + t) - (\frac{1}{|\Sigma|} + t)\log(|\Sigma| - 1))} < 1$$

Clearly $|S|\nu m \leq \sum_{i \in [n]} |R_i|$. So there exists a $j \in [m]$ belonging to at least $\nu|S|$ of the $R_i$ sets. Theorem 4.9 proves that $\nu|S| \leq \frac{\log|\Sigma|}{\log|\Sigma| - H(\frac{1}{|\Sigma|}+t) - (\frac{1}{|\Sigma|}+t)\log(|\Sigma|-1)}$. Therefore,

$|S| \leq \frac{1}{\nu} \frac{\log |\Sigma|}{\log |\Sigma| - H(\frac{1}{|\Sigma|}+t) - (\frac{1}{|\Sigma|}+t)\log(|\Sigma|-1)} = .99n < n$. So $\bar{S}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S}$. That is, $|R_1| < \nu m$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Define $\gamma \triangleq \frac{|[m] \setminus R_1|}{m}$ and $\beta \triangleq \min(\frac{\delta - \nu}{\gamma}, \frac{|\Sigma|-1}{|\Sigma|})$. Let $A$ be a $q$ query algorithm for $\mathbf{C}$ subjected to $\delta$ fraction of the codeword corrupted. Let us consider the probability of error of the decoder over uniformly random $x \in \Sigma^n$, uniformly random $B_1 \subset [m] \setminus R_1$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1$ independently.) The adversary replaces each position in $B_1$ with a value from $\Sigma$ uniformly at random. It replaces each position in $B_2$ with a value from $\Sigma$ uniformly at random, except different from what that position was uncorrupted. Each random corruption is chosen independently. For convenience, define $B \triangleq B_1 \cup B_2$. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1$.

Let us take a moment to bound $\nu$. When $|\Sigma| = 2$,

$$\begin{aligned} v &\triangleq \frac{1}{.99n(1 - H(\frac{1}{2} + t))} \\ &\leq \frac{1}{.99n(1 - (1 - \frac{t^2}{2}))} \qquad \text{Taylor series expansion for } H \\ &= \frac{2}{.99\sqrt{n}} \\ &< \frac{3}{\sqrt{n}} \end{aligned}$$

When $|\Sigma| \geq 3$,

$$\begin{aligned} v &\triangleq \frac{\log|\Sigma|}{.99n(\log|\Sigma| - H(\frac{1}{|\Sigma|} + t) - (\frac{1}{|\Sigma|} + t)\log(|\Sigma| - 1))} \\ &\leq \frac{\log|\Sigma|}{.99n(\log|\Sigma| - 1 - .34\log|\Sigma|)} \qquad n \text{ is large enough so } t \text{ is small enough} \\ &\leq \frac{3}{\sqrt{n}} \qquad \text{for large enough } n \end{aligned}$$

If $A$ linearly decodes using 0 positions, its correctness will be $\frac{1}{|\Sigma|}$, as $x_1$ is uniform in $\Sigma$. If $A$ linearly decodes using 1 position, then we know its correctness cannot be more than $\frac{1}{|\Sigma|} + t$. Also, applying Claim 5.3.1, we see the lowest error $A$ can achieve when linear decoding using at least 2 queries is by linearly decoding using exactly 2 queries. Because of the above, $\nu \leq \frac{3}{\sqrt{n}}$. Using $\delta - \frac{3}{\sqrt{n}}$ (which is less than $\beta$) in place of $\delta$ in the bound of Claim 5.3.1, and noting that that bound is increasing in $\delta$, the result follows. ■

A special case of linear decoders is what is called a matching sum decoder. They were introduced formally in Woodruff [46]:

**Definition 5.3.2** *A MATCHING SUM DECODER is an algorithm such that 1) for a given position of the input the algorithm is tasked to find, the algorithm only queries subsets that form a matching over the codeword positions; and 2) after choosing a subset to query, the algorithm linearly decodes.*

Here is a bound we can state on the correctness of matching sum decoders:

**Claim 5.3.4** *Let* $\mathbf{C} : F^n \to F^m$ *be a code. If* $A$ *is a matching sum decoder operating on* $\mathbf{C}$ *and using exactly* $q$ *positions, then* $\zeta^\star_\delta(A) \leq 1 - q\delta$.

**Proof:** Consider an adversary that finds all the (nonintersecting) subsets of $[m]$ which $A$ queries with nonzero probability. Because each such query set has size $q$, there are at most $m/q$ of them. The adversary takes a node cover of these query sets and, uniformly at random, corrupts $\delta m$ of them. Thus, the algorithm errs with probability at least $\frac{\delta m}{m/q} = q\delta$. ■

Summarizing, one of the main points of this section is to show that, despite how common linear decoder constructions are in the literature, they have serious limitations on their correctness. In fact, the correctness of a linear decoder using exactly $q+1$ positions is worse than our bound on the correctness of a linear decoder using exactly $q$ positions. This is very strange, because intuitively, the correctness of a decoder should get better as the algorithm is allowed the freedom of using more queries. We will show the existence of

algorithms later that have lower correctness. These algorithms perform non-linear operations on the values they receive.

# Chapter 6

# Length Lower Bounds for Three Query LDCs

In this chapter, we prove that three query LDCs that have high enough correctness, compared to the amount of corruption the adversary is allowed to apply, must have exponential size. We prove this first for binary, linear LDCs, then generalize for all binary LDCs, and then generalize for all linear LDCs. It is interesting to note that each one of our length lower bounds uses a two query combinatorial theorem.

## 6.1 Three Query, Binary, Linear LDCs

We use a classic result about two query codes from Goldreich et al. [24].

**Theorem 6.1.1** *([24].) Let $a_1$, ... $a_m$ be a sequence of (not necessarily distinct) elements of $\{0,1\}^n$ such that for every $i \in [n]$ there is a set $M_i$ of disjoint pairs of indices $\{j_1, j_2\}$ such that $e_i = a_{j_1} \oplus a_{j_2}$. Then $m \geq 2^{2\alpha n}$, where $\alpha \triangleq \frac{\sum_{i=1}^{n} |M_i|}{nm}$.*

**Theorem 6.1.2** *Let $\mathbf{C}$: $\{0,1\}^n \rightarrow \{0,1\}^m$ be a linear $(q = 3, \delta, \epsilon)$-LDC with constant $\epsilon > 0$ and $n$ large enough. Then, $m \geq 2^{1.8\alpha n}$ where $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{36}{n})^{1/3} - \nu$ and $\nu \triangleq \frac{10}{n}$.*

**Note:** $\alpha > 0$ when $\frac{1}{2}+\epsilon > 1-3\delta(1-\delta)^2-\delta^3+\phi(n)$ with $\phi(n) = 4((\frac{36}{n})^{1/3}+\nu)$. When $\frac{1}{2}+\epsilon > \mu+1-3\delta(1-\delta)^2-\delta^3+\phi(n)$ for some $\mu \geq 0$, then Fact 6.1.4 implies $\frac{1}{2}+\epsilon > 1-3(\delta-\frac{\mu}{4})(1-(\delta-\frac{\mu}{4}))^2-(\delta-\frac{\mu}{4})^3+\phi(n)$. Therefore, $\alpha > \frac{\mu}{4}$.

**Proof:** Claim 2.2 says that, without loss of generality, we can assume $\mathbf{C}$ has no codeword position that is identically zero.

For $i \in [n]$, define:

$$R_i \triangleq \Big\{ j \in [m] \mid \mathbf{C}(x)_j = x_i \Big\}$$

Also, for each $i \in [n]$, let $M_i$ be a largest matching of edges $\{j_1, j_2\} \subset [m]$ such that $\mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2} = x_i$. For emphasis, no two edges in $M_i$ can intersect because it is a matching. Let $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{36}{n})^{1/3} - \nu$, where $\nu \triangleq \frac{10}{n}$. We will see the rationale for the choice of $\alpha$ at the end of the proof. If $\alpha \leq 0$: because $\epsilon > 0$ requires $m \geq 1$, we are done. So assume $\alpha > 0$. Now consider:

$$S_1 \triangleq \Big\{ i \in [n] \mid |R_i| \geq \nu m \Big\}$$
$$S_2 \triangleq \Big\{ i \in [n] \mid |M_i| \geq \alpha m \Big\}$$

If $|S_2| \geq .9n$, then we can use Theorem 6.1.1 to conclude $m \geq 2^{2*.9\alpha n}$. If $|S_2| < .9n$, then because we know that $|S_1| \leq \frac{m}{\nu m} = .1n$, $\bar{S_1} \cap \bar{S_2}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S_1} \cap \bar{S_2}$. That is, $|R_1| < \nu m$ and $|M_1| < \alpha m$. Consider what happens when the recovery algorithm is tasked to

find $x_1$.

We now construct a node cover $\hat{M}_1 \subset [m]$ of those edges $\{j_1, j_2\} \subset [m]$ such that $\mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2} = x_i$. The graph formed by these edges a union of complete bipartite subgraphs. In each complete bipartite subgraph, the vertices on one of its sides all correspond to the same vector in $\{0,1\}^n$ – call it $a$. The vertices on the other side all correspond to $a + e_1$. Therefore, to construct a small node cover, we can take the union of the vertices of the smaller side of each complete bipartite subgraph. Now, since $M_1$ is a largest matching of these edges, it has one member for each vertex on the smaller side of each complete bipartite subgraph, as well. Therefore, $|\hat{M}_1| = |M_1|$.

Define $\gamma \triangleq \frac{|[m] \setminus (R_1 \cup \hat{M}_1)|}{m}$ and $\beta \triangleq \min(\frac{\delta - \alpha - \nu}{\gamma}, \frac{1}{2})$. Let $A$ be a $(q, \delta, \epsilon)$ algorithm for $\mathbf{C}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus (R_1 \cup \hat{M}_1)$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1 \cup \hat{M}_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1 \cup \hat{M}_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1 \cup \hat{M}_1$.

Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] = \sum_{Q \subset [m], |Q| \leq 3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$. Define $Err_Q \triangleq$
$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound $Err_Q$ depending on all the different possibilities for $Q$ for which $\Pr[A \text{ queries } Q] > 0$.

- $x_1 \notin \text{span}(Q)$: By Theorem 5.1.1, $Err_Q \geq \frac{1}{2}$.
- $|Q| \leq 2$: $e_1 \in \text{span}(Q)$ implies that at least one bit in $Q$ is in $R_1 \cup \hat{M}_1$. But we have already said that $A$ never queries $R_1 \cup \hat{M}_1$.
- $|Q| = 3$ and $e_1 \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$: We can decompose $Err_Q$ into

$$\begin{aligned} Err_Q &= \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k] \cdot \\ &\qquad \Pr_B[|B \cap Q| = k \mid A \text{ queries } Q] \\ &= \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k] \cdot \\ &\qquad \Pr_B[|B \cap Q| = k] \end{aligned}$$

  Note that for any $Q$ and $0 \leq k \leq 3$, the events $A$ queries $Q$ and $|B \cap Q| = k$ are independent. So for any $Q$ and $0 \leq k \leq 3$, $\Pr[A \text{ queries } Q; |B \cap Q| = k] > 0$. Thus, above we are conditioning on events with nonzero

probability. The second equality above also holds because of the independence of $A$ queries $Q$ and $|B \cap Q| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k]$. We can further decompose $Err_{Q,k}$ into

$$Err_{Q,k} =$$
$$\sum_{a,b,c} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abc] \cdot$$
$$\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; |B \cap Q| = k]$$

For simplicity, let us define:

$$q_{abc}^{Q,k} \triangleq \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; |B \cap Q| = k]$$

$e_1$ is not in the span of any two of the query bits taken by themselves, because otherwise, at least one bit would be in $R_1 \cup \hat{M}_1$, and that would violate our assumption on $A$. Thus, the sum of the three bits (when they are uncorrupted) is $x_1$. So $a + b + c = x_1 + (k \bmod 2)$, and the above becomes:

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b+c=k \bmod 2}} q_{abc}^{Q,k} \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abc]$$
$$+ \sum_{\substack{a,b,c \\ a+b+c=1+k \bmod 2}} q_{abc}^{Q,k} \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abc]$$

The event $|B \cap Q| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$\begin{aligned} Err_{Q,k} = & \sum_{\substack{a,b,c \\ a+b+c=k \bmod 2}} q_{abc}^{Q,k} \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \\ & + \sum_{\substack{a,b,c \\ a+b+c=1+k \bmod 2}} q_{abc}^{Q,k} \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \end{aligned}$$

This motivates us to make the following definition. For $a, b, c \in \{0, 1\}$,

$$p_{abc}^{Q} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]$$

This means,

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b+c=k \bmod 2}} (1 - p_{abc}^{Q}) q_{abc}^{Q,k} + \sum_{\substack{a,b,c \\ a+b+c=1+k \bmod 2}} p_{abc}^{Q} q_{abc}^{Q,k}$$

No two query bits are equal, because otherwise, the third one would be in $R_1$, which would also violate our assumption on $A$. Since the sum of the three bits is $e_1$, the three bits are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 $(\mathbf{C}(x)+B)_{j_1}$, $(\mathbf{C}(x)+B)_{j_2}$, and $(\mathbf{C}(x)+B)_{j_3}$ are three independent, uniformly random bits. Thus, $\forall k, a, b, c$: $q_{abc}^{Q,k} = \frac{1}{8}$. So, when $k$ is even,

$$\begin{aligned} Err_{Q,k} = \Big( & (1 - p_{000}^{Q}) + (1 - p_{011}^{Q}) + (1 - p_{110}^{Q}) + (1 - p_{101}^{Q}) + \\ & p_{100}^{Q} + p_{010}^{Q} + p_{001}^{Q} + p_{111}^{Q} \Big) / 8 \end{aligned}$$

For simplicity, define $P_Q \triangleq \Big((1-p^Q_{000})+(1-p^Q_{011})+(1-p^Q_{110})+(1-p^Q_{101})+p^Q_{100}+p^Q_{010}+p^Q_{001}+p^Q_{111}\Big)/8$. On the other hand, when $k$ is odd,

$$\begin{aligned} Err_{Q,k} &= \Big(p^Q_{000}+p^Q_{011}+p^Q_{110}+p^Q_{101}+ \\ &\qquad (1-p^Q_{100})+(1-p^Q_{010})+(1-p^Q_{001})+(1-p^Q_{111})\Big)/8 \\ &= 1-P_Q \end{aligned}$$

By Lemma 5.2.2, we know the following two things:

$$\begin{aligned} &\Pr_B[|B\cap Q|=0]+\Pr_B[|B\cap Q|=2] \\ &\qquad \geq (1-\beta)^3-\frac{9}{\gamma m}+3\beta^2(1-\beta)-\frac{9}{\gamma m} \\ &\qquad \geq (1-\beta)^3+3\beta^2(1-\beta)-\frac{18}{\gamma m} \\ &\qquad \geq (1-\beta)^3+3\beta^2(1-\beta)-\frac{36}{m} \\ &\Pr_B[|B\cap Q|=1]+\Pr_B[|B\cap Q|=3] \\ &\qquad \geq 3\beta(1-\beta)^2-\frac{9}{\gamma m}+\beta^3-\frac{9}{\gamma m} \\ &\qquad \geq 3\beta(1-\beta)^2+\beta^3-\frac{36}{m} \end{aligned}$$

Combining everything, we find

$$\begin{aligned} Err_Q &\geq \Big((1-\beta)^3+3\beta^2(1-\beta)-\frac{36}{m}\Big)P_Q+ \\ &\qquad \Big(3\beta(1-\beta)^2+\beta^3-\frac{36}{m}\Big)(1-P_Q) \\ &= 3\beta(1-\beta)^2+\beta^3-\frac{36}{m}+ \\ &\qquad \Big((1-\beta)^3+3\beta^2(1-\beta)-3\beta(1-\beta)^2-\beta^3\Big)P_Q \\ &= 3\beta(1-\beta)^2+\beta^3-\frac{36}{m}+\Big((1-\beta)-\beta\Big)^3P_Q \end{aligned}$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}$.

Since for all $Q$, $Err_Q \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}$, where the probability is only over the internal coin flips of $A$.

Remember that $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$. When $\beta = \frac{\delta-\alpha-\nu}{\gamma}$, first note that the expression $3\beta(1-\beta)^2 + \beta^3$ is strictly increasing in $\beta$. Therefore, we can lower bound $3\beta(1-\beta)^2+\beta^3-\frac{36}{m}$ evaluated at $\beta = \frac{\delta-\alpha-\nu}{\gamma}$ with $3\hat{\beta}(1-\hat{\beta})^2+\hat{\beta}^3-\frac{36}{m}$ evaluated at $\hat{\beta} = \delta - \alpha - \nu = \frac{1}{2} + (\frac{36}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}$:

$$3\Big(\frac{1}{2} + (\frac{36}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}\Big)\Big(\frac{1}{2} - (\frac{36}{n})^{1/3} + (\frac{\epsilon}{4})^{1/3}\Big)^2 + \Big(\frac{1}{2} + (\frac{36}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}\Big)^3 - \frac{36}{m}$$

Because of Fact 6.1.3, this expression is lower bounded by

$$\geq 3\Big(\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}\Big)\Big(\frac{1}{2} + (\frac{\epsilon}{4})^{1/3}\Big)^2 + \Big(\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}\Big)^3 + \frac{36}{n} - \frac{36}{m}$$
$$\geq \frac{1}{2} - \epsilon + \frac{36}{n} - \frac{36}{m}$$

The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Therefore, this expression is more than $\frac{1}{2} - \epsilon$. (It is apparent now that the choice of $\alpha$ at the start of the proof was made so that $\xi = \delta - \alpha - \nu - (\frac{36}{n})^{1/3}$ would be the solution to $3\xi(1-\xi)^2 + \xi^3 = \frac{1}{2} - \epsilon$.)

When $\beta = \frac{1}{2}$, $3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} = \frac{1}{2} - \frac{36}{m} > \frac{1}{2} - \epsilon$ for large enough $n$ (again, note that $m > n$).

So, we have shown there is a situation in which the error is more than $\frac{1}{2} - \epsilon$. But this contradicts the fact that the recovery algorithm achieves $(3, \delta, \epsilon)$ on $\mathbf{C}$. ■

We note that this theorem can be extended for all $\delta$ and sub-constant $\epsilon$. We make no restrictions on $\delta$ in the proof, although, for the bound to be non-trivial, we need $\delta > \Omega(\frac{1}{n^{1/3}})$ (because $\epsilon \leq \frac{1}{2}$). We do make a restriction of $\epsilon$ in the proof – we need $\epsilon > \Omega(\frac{1}{n})$. This all means that when $\delta$ is small, the range of usefulness of this theorem $\frac{1}{2} + \epsilon > 1 - 3\delta(1-\delta)^2 - \delta^3 + O(\frac{1}{n^{1/3}})$ is very close to the $\delta$ versus $\epsilon$ tradeoff of the best known three query construction from Woodruff [46]: $\frac{1}{2} + \epsilon = 1 - 3\delta - \eta$, where $\eta > 0$ is arbitrarily small.

As an addendum, here are the small facts used above.

**Fact 6.1.3** *Let* $Z(\beta) \triangleq 3\beta(1-\beta)^2 + \beta^3$. *For any* $\beta$ *and* $\rho \geq 0$, $Z(\beta + \rho) \geq Z(\beta) + \rho^3$.

**Proof:** Let $x \triangleq \beta + \rho/2$ and $p \triangleq \rho/2$. So now we need to lower bound $Z(x+p) - Z(x-p)$. Note that $Z(x)$ can be expressed as $3x - 6x^2 + 4x^3$.

Therefore,

$$
\begin{aligned}
& Z(x+p) - Z(x-p) \\
&= \Big(3(x+p) - 6(x+p)^2 + 4(x+p)^3\Big) - \Big(3(x-p) - 6(x-p)^2 + 4(x-p)^3\Big) \\
&= 3\Big((x+p) - (x-p)\Big) - 6\Big((x+p)^2 - (x-p)^2\Big) + 4\Big((x+p)^3 - (x-p)^3\Big) \\
&= 3(2p) - 6(4xp) + 4(6x^2p + 2p^3) \\
&= 6p - 24xp + 24x^2p + 8p^3 \\
&= 3\rho - 12x\rho + 12x^2\rho + \rho^3 \\
&= 3\rho(1-2x)^2 + \rho^3 \\
&\geq \rho^3
\end{aligned}
$$

■

**Fact 6.1.4** *Let* $Z(\beta) \triangleq 3\beta(1-\beta)^2 + \beta^3$. *For any* $\beta$ *and* $0 \leq \rho \leq 1$ *such that* $0 \leq \beta + \frac{\rho}{2} \leq 1$, $Z(\beta+\rho) \leq Z(\beta) + 4\rho$.

**Proof:** We can use the same notation and the same first several steps of the Fact 6.1.3 to get

$$Z(x+p) - Z(x-p) = 3\rho(1-2x)^2 + \rho^3 \leq 3\rho + \rho^3 \leq 4\rho$$

■

## 6.2 Three Query, Binary, Possibly Non-Linear LDCs

In this section, we extend our lower bound on the length of three query linear LDCs to the possibly non-linear case. We use a combinatorial lemma from Ben-Aroya et al. [8]. This lemma is the foundation for the first classical proof that non-linear, two query LDCs must have exponential length. Recall that this result was first obtained using quantum computing methods by Kerenidis and de Wolf [34]. Here is the combinatorial lemma, implicit from Theorem 11 in [8].

**Theorem 6.2.1** *([8].) Let $0 \leq \epsilon, c \leq \frac{1}{2}$ be constants. Let $a_1$, ... $a_m$ be a sequence of (not necessarily distinct) $\{0,1\}^n \to \{0,1\}$ functions such that for every $i \in [n]$ there is a set $M_i$ of disjoint pairs ($|M_i| \geq cm$) of indices $\{j_1, j_2\}$ such that*

$$\left| \Pr_{x \in \{0,1\}^n}[x = a_{j_1}(x) + a_{j_2}(x)] - \Pr_{x \in \{0,1\}^n}[x \neq a_{j_1}(x) + a_{j_2}(x)] \right| \geq \epsilon$$

*Then $m \geq 2^{c^2\epsilon^2 n}$.*

Now here is our new three query, binary, possibly non-linear lower bound.

**Theorem 6.2.2** *Fix $t_1 > 0$ and $t_2 > 0$. Let $\mathbf{C}$: $\{0,1\}^n \to \{0,1\}^m$ be a $(q = 3, \delta, \epsilon)$-LDC with constant $\epsilon > \frac{3}{2}(t_1 + t_2)$ and $n$ large enough. Let $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{3}{2}(t_1 + t_2) + \frac{36}{n})^{1/3} - \nu \geq 0$ and $\nu \triangleq \frac{10}{n(1 - H(\frac{1}{2} + \frac{t_1}{2}))}$. If $\alpha > 0$, then $m \geq 2^{.225\alpha^2 t_2^2 n}$.*

**Note 1:** $\alpha > 0$ when $\frac{1}{2} + \epsilon > 1 - 3\delta(1-\delta)^2 - \delta^3 + \phi(n)$ with $\phi(n) = 4((\frac{3}{2}(t_1+t_2)+\frac{36}{n})^{1/3}+\nu)$. When $\frac{1}{2}+\epsilon > \mu+1-3\delta(1-\delta)^2-\delta^3+\phi(n)$ for some $\mu \geq 0$, then Fact 6.1.4 implies $\frac{1}{2}+\epsilon > 1-3(\delta-\frac{\mu}{4})(1-(\delta-\frac{\mu}{4}))^2-(\delta-\frac{\mu}{4})^3+\phi(n)$. Therefore, $\alpha > \frac{\mu}{4}$.

**Note 2:** An appealing combination of $t_1$ and $t_2$ ($t_1 = t_2 = \frac{1}{n^{1/3}}$) gives the following bound: Let $\mathbf{C}$: $\{0,1\}^n \to \{0,1\}^m$ be a $(q = 3, \delta, \epsilon)$-LDC with constant $\epsilon > \frac{3}{n^{1/3}}$ and $n$ large enough. Let $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{3}{n^{1/3}} + \frac{36}{n})^{1/3} - \nu \geq 0$ and $\nu \triangleq \frac{10}{n(1-H(\frac{1}{2}+\frac{1}{2n^{1/3}}))}$. If $\alpha > 0$, then $m \geq 2^{.225\alpha^2 n^{1/3}}$.

**Proof:** For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \left| \Pr_{x \in \{0,1\}^n}[x_i = \mathbf{C}(x)_j] - \Pr_{x \in \{0,1\}^n}[x_i \neq \mathbf{C}(x)_j] \right| \geq t_1 \right\}$$

Also, for each $i \in [n]$, let $M_i$ be a largest matching of edges $\{j_1, j_2\} \subset [m]$ such that

$$\left| \Pr_{x \in \{0,1\}^n}[x_i = \mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2}] - \Pr_{x \in \{0,1\}^n}[x_i \neq \mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2}] \right| \geq t_2$$

For emphasis, no two edges in $M_i$ can intersect because it is a matching. Let $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{3}{2}(t_1+t_2) + \frac{36}{n})^{1/3} - \nu$ where $\nu \triangleq \frac{10}{n(1-H(\frac{1}{2}+\frac{t_1}{2}))}$. We will see the rationale for this choice at the end of the proof. Now consider:

$$S_1 \triangleq \left\{ i \in [n] \mid |R_i| \geq \nu m \right\}$$
$$S_2 \triangleq \left\{ i \in [n] \mid |M_i| \geq \frac{\alpha}{2} m \right\}$$

If $|S_2| \geq .9n$, then we can use Theorem 6.1.1 to conclude $m \geq 2^{\frac{.9}{4}\alpha^2 t_2^2 n}$. If $|S_2| < .9n$, then let us consider the following. Clearly $|S_1|\nu m \leq \sum_{i\in[n]} |R_i|$. So there exists a $j \in [m]$ belonging to at least $\nu|S_1|$ of the $R_i$ sets. Theorem 2 from [32] then proves that $\nu|S_1| \leq \frac{1}{1-H(\frac{1}{2}+\frac{t_1}{2})}$. Therefore, $|S_1| \leq \frac{1}{\nu}\frac{1}{1-H(\frac{1}{2}+\frac{t_1}{2})} = .1n$. So $\bar{S_1} \cap \bar{S_2}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S_1} \cap \bar{S_2}$. That is, $|R_1| < \nu m$ and $|M_1| < \frac{\alpha}{2}m$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries three positions. If the algorithm ever queried fewer than three positions, have it query more and ignore the additional values obtained.

Construct $\hat{M_1} \subset [m]$ as the union of all the members in $M_1$. So $|\hat{M_1}| = 2|M_1|$.

Define $\gamma \triangleq \frac{|[m]\setminus(R_1 \cup \hat{M_1})|}{m}$ and $\beta \triangleq \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$. Let $A$ be a $(q, \delta, \epsilon)$ algorithm for $\mathbf{C}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus (R_1 \cup \hat{M_1})$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1 \cup \hat{M_1}$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1 \cup \hat{M_1}$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote

the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1 \cup \hat{M}_1$.

Without loss of generality, we assume that $A$ flips all of its random coins first, and then, based on those random values, chooses a query set $Q \subset [m]$ and a deterministic function $\phi$ to apply on the three values it receives from querying $Q$. Without loss of generality, $Q = \{1, 2, 3\}$. We use the shorthand "$Q$, $\phi$" to mean the event $A$ has chosen to query $Q$ and use function $\phi$. Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] = \sum_{Q \subset [m]:|Q|=3,\phi} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid Q, \phi] \Pr[Q, \phi]$$

Define $Err_{Q,\phi} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid Q, \phi]$. We will bound $Err_{Q,\phi}$ using the following concept. Define the correlation between two Boolean functions $f$ and $g$ as

$$Corr(f, g) \triangleq \Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)]$$

Let us consider the quantity $|Corr(x_i, \phi(Y_1, Y_2, Y_3))|$. With $\chi_S(Y_1, Y_2, Y_3) \triangleq \sum_{s \in S} Y_s$ for $S \subseteq \{1, 2, 3\}$, Lemma 6.2.3 at the end of this

proof gives

$$\begin{aligned}
&\Big|Corr(x_i, \phi(Y_1, Y_2, Y_3))\Big| \\
&\quad \leq \Big|Corr(x_i, 0)\Big| + \sum_{S \subseteq \{1,2,3\} \,:\, |S|=1} \Big|Corr(x_i, \chi_S(Y_1, Y_2, Y_3))\Big| + \\
&\quad \sum_{S \subseteq \{1,2,3\} \,:\, |S|=2} \Big|Corr(x_i, \chi_S(Y_1, Y_2, Y_3))\Big| + \Big|Corr(x_i, Y_1 + Y_2 + Y_3)\Big|
\end{aligned}$$

The first term of this expression is 0 because $\Pr_x[x_i = 0] = \frac{1}{2}$. The three absolute values in the second term are each at most $t_1$. This is because for any $j \in [m]$, if $|Corr(x_i, \mathbf{C}(x)_j)| > t_1$, then $j_1$ is corrupted by $B$ into a uniformly random value in $\{0, 1\}$. Therefore, the correlation of the corrupted value with $x_i$ is 0. Similarly, the three absolute values in the third term above are each at most $t_2$. (Note that if two members of $\{1, 2, 3\}$, for instance $\{1, 2\}$, have $|Corr(x_i, \mathbf{C}(x)_1 + \mathbf{C}(x)_2)| > t_2$ then at least one of $\{1, 2\}$ must be in $\hat{M}_i$ and so are corrupted, or else $M_i$ would not be a maximum matching.) This gives:

$$\Big|Corr(x_i, \phi(Y_1, Y_2, Y_3))\Big| \leq 3t_1 + 3t_2 + \Big|Corr(x_i, Y_1 + Y_2 + Y_3)\Big|$$

For simplicity, let us temporarily just operate on $|Corr(x_i, Y_1+Y_2+Y_3)|$:

$$\begin{aligned}
&\Big|Corr(x_i, Y_1 + Y_2 + Y_3)\Big| \\
&\qquad = \Big|\Pr[x_i = Y_1 + Y_2 + Y_3] - \Pr[x_i \neq Y_1 + Y_2 + Y_3]\Big| \\
&\qquad = \Big|\Pr[0 = x_i + Y_1 + Y_2 + Y_3] - \Pr[0 \neq x_i + Y_1 + Y_2 + Y_3]\Big|
\end{aligned}$$

Because of the independence of $x$ and $B$, we have:

$$
\begin{aligned}
&= \Big| \Pr[0 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3] \Pr[0 = B_1 + B_2 + B_3] + \\
&\qquad \Pr[1 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3] \Pr[1 = B_1 + B_2 + B_3] - \\
&\qquad \Pr[0 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3] \Pr[1 = B_1 + B_2 + B_3] - \\
&\qquad \Pr[1 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3] \Pr[0 = B_1 + B_2 + B_3] \Big| \\
&= \Big| (\Pr[0 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3] - \\
&\qquad\qquad \Pr[1 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3]) \\
&\qquad (\Pr[0 = B_1 + B_2 + B_3] - \Pr[1 = B_1 + B_2 + B_3]) \Big| \\
&= \Big| Corr(x_i, \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3) Corr(0, B_1 + B_2 + B_3) \Big| \\
&\leq \Big| Corr(x_i, \mathbf{C}(x)_1 + \mathbf{C}(x)_2 + \mathbf{C}(x)_3) \Big| \Big| Corr(0, B_1 + B_2 + B_3) \Big| \\
&\leq \Big| Corr(0, B_1 + B_2 + B_3) \Big|
\end{aligned}
$$

Each member of $B$ has been corrupted with probability at least $\beta$. By Lemma 5.2.2, we know:

$$
\begin{aligned}
\Pr_B[|B \cap Q| = 1] + \Pr_B[|B \cap Q| = 3] &\geq 3\beta(1-\beta)^2 - \frac{9}{\gamma m} + \beta^3 - \frac{9}{\gamma m} \\
&\geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}
\end{aligned}
$$

Because $\beta \leq \frac{1}{2}$, we have

$$\begin{aligned}
&\Big|Corr(x_i, \phi(Y_1, Y_2, Y_3))\Big| \\
&\qquad \leq 3t_1 + 3t_2 + \Big((1 - \Pr_B[|B \cap Q| = 1] - \Pr_B[|B \cap Q| = 3]\Big) - \\
&\qquad\qquad \Big(\Pr_B[|B \cap Q| = 1] + \Pr_B[|B \cap Q| = 3])\Big) \\
&\qquad = 3t_1 + 3t_2 + 1 - 2\Big(\Pr_B[|B \cap Q| = 1] + \Pr_B[|B \cap Q| = 3]\Big) \\
&\qquad \leq 3t_1 + 3t_2 + 1 - 2\Big(3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}\Big)
\end{aligned}$$

Noting that $Err_{Q,\phi} \leq \frac{1}{2}$ or else the algorithm would just guess randomly, we have:

$$\begin{aligned}
(1 - Err_{Q,\phi}) - Err_{Q,\phi} &= \Big|(1 - Err_{Q,\phi}) - Err_{Q,\phi}\Big| = \Big|Corr(x_i, \phi(Y_1, Y_2, Y_3))\Big| \\
&\Rightarrow Err_{Q,\phi} \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{3}{2}(t_1 + t_2) - \frac{36}{m}
\end{aligned}$$

Since for all $Q$ and $\phi$, $Err_{Q,\phi} \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{3}{2}(t_1 + t_2) - \frac{36}{m}$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{3}{2}(t_1 + t_2) - \frac{36}{m}$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{3}{2}(t_1 + t_2) - \frac{36}{m}$, where the probability is only over the internal coin flips of $A$.

Remember that $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$. When $\beta = \frac{\delta-\alpha-\nu}{\gamma}$, first note that the expression $3\beta(1-\beta)^2 + \beta^3$ is strictly increasing in $\beta$. Therefore, we can lower bound $3\beta(1-\beta)^2 + \beta^3 - \frac{3}{2}(t_1 + t_2) - \frac{36}{m}$ evaluated at $\beta = \frac{\delta-\alpha-\nu}{\gamma}$ with $3\hat{\beta}(1-\hat{\beta})^2 + \hat{\beta}^3 - \frac{3}{2}(t_1 + t_2) - \frac{36}{m}$ evaluated at $\hat{\beta} = \delta - \alpha - \nu = \frac{1}{2} + (\frac{3}{2}(t_1 +$

$t_2) + \frac{36}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}$:

$$3\Big(\frac{1}{2} + (\frac{3}{2}(t_1+t_2) + \frac{36}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}\Big)\Big(\frac{1}{2} - (\frac{3}{2}(t_1+t_2) + \frac{36}{n})^{1/3} + (\frac{\epsilon}{4})^{1/3}\Big)^2 + \Big(\frac{1}{2} + (\frac{3}{2}(t_1+t_2) + \frac{36}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}\Big)^3 - \frac{3}{2}(t_1+t_2) - \frac{36}{m}$$

Because of Fact 6.1.3, this expression is lower bounded by

$$\geq 3\Big(\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}\Big)\Big(\frac{1}{2} + (\frac{\epsilon}{4})^{1/3}\Big)^2 + \Big(\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}\Big)^3 + \frac{3}{2}(t_1+t_2) + \frac{36}{n} - \frac{3}{2}(t_1+t_2) - \frac{36}{m}$$

The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Therefore, this expression is more than $\frac{1}{2} - \epsilon$. (It is apparent now that the choice of $\alpha$ at the start of the proof was made so that $\xi = \delta - \alpha - \nu - (\frac{3}{2}(t_1+t_2) + \frac{36}{n})^{1/3}$ would be the solution to $3\xi(1-\xi)^2 + \xi^3 = \frac{1}{2} - \epsilon$.)

When $\beta = \frac{1}{2}$, $3\beta(1-\beta)^2 + \beta^3 - \frac{3}{2}(t_1+t_2) - \frac{36}{m} = \frac{1}{2} - \frac{3}{2}(t_1+t_2) - \frac{36}{m} > \frac{1}{2} - \epsilon$ for large enough $n$ (again, note that $m > n$) and $\epsilon > \frac{3}{2}(t_1+t_2)$.

So, we have shown there is a situation in which the error is more than $\frac{1}{2} - \epsilon$. But this contradicts the fact that the recovery algorithm achieves $(3, \delta, \epsilon)$ on $\mathbf{C}$. ■

**Lemma 6.2.3** *Define* $\chi_S(Y_1, Y_2, Y_3) \triangleq \sum_{s \in S} Y_s$. *For any Boolean functions* $f$

*and* $g(Y_1, Y_2, Y_3)$,

$$\Big|Corr(f,g)\Big| \leq \sum_{S\subseteq\{1,2,3\}} \Big|Corr(f, \chi_S(Y_1,Y_2,Y_3))\Big|$$

**Proof:** Since $f$ and $g$ are Boolean, we can express them using the $\{-1,1\}$ notation: $f^*$ will be $\{-1,1\}^n \rightarrow \{-1,1\}$ for some $n$, where $f^*(x_1,x_2,...,x_n) \triangleq (-1)^{f(\frac{1-x_1}{2},\frac{1-x_2}{2},...\frac{1-x_n}{2})}$, and $g^*$ will be defined analogously from $g$. Using this notation,

$$\begin{aligned} Corr(f,g) &= Corr(f^*,g^*) \\ &= \sum_{X\in\{-1,1\}^n,\ Y_1,Y_2,Y_3\in\{-1,1\}} \Pr(X,Y_1,Y_2,Y_3) f^*(X) g^*(Y_1,Y_2,Y_3) \end{aligned}$$

where $X$ is the input of $f^*$. The Fourier decomposition of $g^*$ is: $g^*(Y_1,Y_2,Y_3) = \sum_{S\subseteq\{1,2,3\}} \hat{g}(S)\chi_S^*(Y_1,Y_2,Y_3)$ where the $\hat{g}(S)$ are constants and $\chi_S^*(Y_1,Y_2,Y_3) \triangleq \prod_{s\in S} Y_s$ for $S \subseteq \{1,2,3\}$. Therefore,

$$\begin{aligned} &Corr(f^*,g^*) = \\ &\sum_{X\in\{-1,1\}^n,\ Y_1,Y_2,Y_3\in\{-1,1\}} \Pr(X,Y_1,Y_2,Y_3) f^*(X) \sum_{S\subseteq\{1,2,3\}} \hat{g}(S)\chi_S^*(Y_1,Y_2,Y_3) \end{aligned}$$

Parseval's theorem provides that $\hat{g}(S) \leq 1$ for all $S$. So,

$$\begin{aligned} &\Big|Corr(f^*,g^*)\Big| \\ &\leq \Big| \sum_{X\in\{-1,1\}^n,\ Y_1,Y_2,Y_3\in\{-1,1\}} \Pr(X,Y_1,Y_2,Y_3) f^*(X) \sum_{S\subseteq\{1,2,3\}} \chi_S^*(Y_1,Y_2,Y_3)\Big| \\ &= \Big| \sum_{S\subseteq\{1,2,3\}} Corr(f^*, \chi_S^*(Y_1,Y_2,Y_3))\Big| \\ &= \Big| \sum_{S\subseteq\{1,2,3\}} Corr(f, \chi_S(Y_1,Y_2,Y_3))\Big| \end{aligned}$$

■

## 6.3 Three Query, Linear LDCs over Any Field

Now we generalize the three query, binary lower bound to three query, linear codes over any field. Instead of using the core combinatorial theorem directly from Goldreich et al. [24], we use an adaptation of it by Dvir and Shpilka [17]:

**Theorem 6.3.1** *([17].) Let $F$ be a field. Let $a_1$, ... $a_m$ be a sequence of (not necessarily distinct) elements of $F^n$ such that for every $i \in [n]$ there is a set $M_i$ of disjoint pairs of indices $\{j_1, j_2\}$ such that $e_i \in \text{span}(a_{j_1}, a_{j_2})$. Then $m \geq 2^{\alpha n - 1}$, where $\alpha \triangleq \frac{\sum_{i=1}^{n} |M_i|}{nm}$.*

**Theorem 6.3.2** *Let $\mathbf{C}$: $F^n \rightarrow F^m$ be a linear $(q = 3, \delta, \epsilon)$-LDC with constant $\epsilon > 0$ and $n$ large enough. Then, $m \geq 2^{.45\alpha n - 1}$ where $\alpha \triangleq \delta + (\frac{|F|\epsilon}{|F|-1})^{1/3} - 1 - (\frac{27|F|}{n})^{1/3} - \nu$ and $\nu \triangleq \frac{10}{n}$.*

**Note:** $\alpha > 0$ when $\epsilon > \frac{|F|-1}{|F|}((1-\delta)^3 + \phi(n))$ with $\phi(n) = 4((\frac{27|F|}{n})^{1/3} + \nu)$. When $\epsilon > \mu + \frac{|F|-1}{|F|}((1-\delta)^3 + \phi(n))$ for some $\mu \geq 0$, then Fact 6.3.4 implies $\epsilon > \frac{|F|-1}{|F|}((1 - \delta + \frac{|F|}{|F|-1}\frac{\mu}{4})^3 + \phi(n))$. Therefore, $\alpha > \frac{|F|}{|F|-1}\frac{\mu}{4}$.

**Proof:** Claim 2.2 says that, without loss of generality, we can assume $\mathbf{C}$ has no codeword position that is identically zero.

For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \exists c \in F^* : \mathbf{C}(x)_j = cx_i \right\}$$

Also, for each $i \in [n]$, let $M_i$ be a largest matching of edges $\{j_1, j_2\} \subset [m]$ such that there exists a non-zero linear combination of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$ equalling $x_i$. For emphasis, no two edges in $M_i$ can intersect because it is a matching. Define $\alpha \triangleq \delta + (\frac{|F|\epsilon}{|F|-1})^{1/3} - 1 - (\frac{27|F|}{n})^{1/3} - \nu$ where $\nu \triangleq \frac{10}{n}$. We will see the rationale for this choice at the end of the proof. If $\alpha \leq 0$: because $\epsilon > 0$ requires $m \geq 1$, we are done. So assume $\alpha > 0$. Now consider:

$$S_1 \triangleq \left\{ i \in [n] \mid |R_i| \geq \nu m \right\}$$
$$S_2 \triangleq \left\{ i \in [n] \mid |M_i| \geq \frac{\alpha}{2} m \right\}$$

If $|S_2| \geq .9n$, then we can use Theorem 6.3.1 to conclude $m \geq 2^{.9*.5\alpha n - 1}$. If $|S_2| < .9n$, then because we know that $|S_1| \leq \frac{m}{\nu m} = .1n$, $\bar{S_1} \cap \bar{S_2}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S_1} \cap \bar{S_2}$. That is, $|R_1| < \nu m$ and $|M_1| < \frac{\alpha}{2} m$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Construct $\hat{M_1} \subset [m]$ as the union of all the edges in $M_1$. Note that $|\hat{M_1}| = 2|M_1|$.

Define $\gamma \triangleq \frac{|[m] \setminus (R_1 \cup \hat{M_1})|}{m}$ and $\beta \triangleq \min(\frac{\delta - \alpha - \nu}{\gamma}, 1)$. Let $A$ be a $(q, \delta, \epsilon)$ algorithm for $\mathbf{C}$. Let us consider the probability of error of the decoder over

uniformly random $x \in F^n$ and the following distribution for the adversary. The adversary will choose $B_1 \subset [m] \setminus (R_1 \cup \hat{M}_1)$ uniformly at random such that $|B_1| = \beta\gamma m$. The adversary will corrupt the codeword $\mathbf{C}(x)$ by adding an independent, uniformly random member of $F$ in the positions present in $B \triangleq B_1 \cup R_1 \cup \hat{M}_1$. For simplicity, let $\mathbf{C}(x) + B$ denote this corrupted codeword. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1 \cup \hat{M}_1$.

Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1]$$
$$= \sum_{Q \subset [m], |Q| \leq 3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$. Define $Err_Q \triangleq$
$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound $Err_Q$ depending on all the different possibilities for $Q$ for which $\Pr[A \text{ queries } Q] > 0$.

- $x_1 \notin \text{span}(Q)$: By Theorem 5.1.1, $Err_Q \geq \frac{1}{|F|}$.
- $|Q| \leq 2$: $e_1 \in \text{span}(Q)$ implies that at least one bit in $Q$ is in $R_1 \cup \hat{M}_1$. But we have already said that $A$ never queries $R_1 \cup \hat{M}_1$.
- $|Q| = 3$ and $e_1 \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$: $e_1$ is not in the span of any two of the bits taken by themselves, because otherwise, at least one bit would

be in $R_1 \cup \hat{M}_1$, and that would violate our assumption on $A$. Thus, there exists a non-zero linear combination of the three bits that equals $x_1$. Let us call it $L(\mathbf{C}(x)_{j_1}, \mathbf{C}(x)_{j_2}, \mathbf{C}(x)_{j_3}) = x_1$.

Use the notation $B \cdot Q$ to mean $L((\mathbf{C}(0)+B)_{j_1}, (\mathbf{C}(0)+B)_{j_2}, (\mathbf{C}(0)+B)_{j_3})$. We can decompose $Err_Q$ into

$$Err_Q = \sum_{k \in F} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k] \cdot \Pr_{B}[B \cdot Q = k \mid A \text{ queries } Q]$$

For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k]$. We can further decompose $Err_{Q,k}$ into

$$Err_{Q,k} = \sum_{a,b,c \in F} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k; (\mathbf{C}(x)+B)_Q = abc] \cdot \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; B \cdot Q = k]$$

For simplicity, let us define:

$$q_{abc}^{Q,k} \triangleq \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; B \cdot Q = k]$$

Note that $L(a,b,c) = x_1 + k$. So we decompose even further. For fixed $a, b, c$,

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k; (\mathbf{C}(x)+B)_Q = abc] = \sum_{r \in F : L(a,b,c) \neq r+k} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; B \cdot Q = k; (\mathbf{C}(x)+B)_Q = abc]$$

The event $B \cdot Q = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$= \sum_{r \in F : L(a,b,c) \neq r+k} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]$$

This motivates us to make the following definition. For $a, b, c \in F$,

$$p^Q_{abc}(r) \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]$$

Where $\sum_{r \in F} p^Q_{abc}(r) = 1$. This means,

$$Err_{Q,k} = \sum_{a,b,c \in F} \sum_{r \in F : L(a,b,c) \neq r+k} p^Q_{abc}(r) q^{Q,k}_{abc}$$

No two bits are equal, because otherwise, either at least one of the three bits would be in $\hat{M}_1$ or the third bit would be in $R_1$, which would also violate our assumption on $A$. Since there is a non-zero linear combination of the bits that equals $e_1$, the three bits are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x)+B)_{j_1}$, $(\mathbf{C}(x)+B)_{j_2}$, and $(\mathbf{C}(x)+B)_{j_3}$ are three independent, uniformly random bits. Thus, $\forall k, a, b, c$: $q^{Q,k}_{abc} = \frac{1}{|F|^3}$. So,

$$\begin{aligned} Err_{Q,k} &= \sum_{a,b,c \in F} \sum_{r \in F : L(a,b,c) \neq r+k} p^Q_{abc}(r) \frac{1}{|F|^3} \\ &= \sum_{a,b,c \in F} \Big(1 - p^Q_{abc}(L(a,b,c) - k)\Big) \frac{1}{|F|^3} \end{aligned}$$

For simplicity, define $P_Q(k) = \sum_{a,b,c \in F} p^Q_{abc}(L(a,b,c) - k)\frac{1}{|F|^3}$. Note that $\sum_{k \in F} P_Q(k) = 1$. Also, $Err_{Q,k} = 1 - P_Q(k)$. Therefore,

$$
\begin{aligned}
Err_Q &= \sum_{k \in F}(1 - P_Q(k)) \Pr_B[B \cdot Q = k \mid A \text{ queries } Q] \\
&= \sum_{k \in F}(1 - P_Q(k))\Big(\Pr_B[B \cdot Q = k \mid |B \cap Q| > 0; A \text{ queries } Q] \cdot \\
&\qquad \Pr_B[|B \cap Q| > 0 \mid A \text{ queries } Q] + \\
&\qquad \Pr_B[B \cdot Q = k \mid |B \cap Q| = 0; A \text{ queries } Q] \cdot \\
&\qquad \Pr_B[|B \cap Q| = 0 \mid A \text{ queries } Q]\Big) \\
&= \sum_{k \in F}(1 - P_Q(k))\Big(\Pr_B[B \cdot Q = k \mid |B \cap Q| > 0; A \text{ queries } Q] \cdot \\
&\qquad \Pr_B[|B \cap Q| > 0] + \\
&\qquad \Pr_B[B \cdot Q = k \mid |B \cap Q| = 0; A \text{ queries } Q] \cdot \\
&\qquad \Pr_B[|B \cap Q| = 0]\Big) \\
&= \sum_{k \in F}(1 - P_Q(k))\frac{1}{|F|}\Big(1 - (1-\beta)^3 - 3\frac{9}{\gamma m}\Big) + \\
&\qquad (1 - P_Q(0))\Big((1-\beta)^3 - \frac{9}{\gamma m}\Big) \\
&= \frac{|F|-1}{|F|}\Big(1 - (1-\beta)^3 - \frac{27}{\gamma m}\Big) + \\
&\geq \frac{|F|-1}{|F|}\Big(1 - (1-\beta)^3 - \frac{27}{\gamma m}\Big) \\
&\geq \frac{|F|-1}{|F|}\Big(1 - (1-\beta)^3 - \frac{27|F|}{m}\Big)
\end{aligned}
$$

We have used that $\gamma m > (1-\delta)m > \frac{m}{|F|}$ in the last line. Since for all $Q$, $Err_Q \geq \frac{|F|-1}{|F|}(1-(1-\beta)^3 - \frac{27|F|}{m})$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq \frac{|F|-1}{|F|}(1-(1-\beta)^3 - \frac{27|F|}{m})$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq$

$\frac{|F|-1}{|F|}(1-(1-\beta)^3-\frac{27|F|}{m})$, where the probability is only over the internal coin flips of $A$.

Remember that $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, 1)$. When $\beta = \frac{\delta-\alpha-\nu}{\gamma}$, first note that the expression $1-(1-\beta)^3$ is strictly increasing in $\beta$. Therefore, we can lower bound $\frac{|F|-1}{|F|}(1-(1-\beta)^3-\frac{27|F|}{m})$ evaluated at $\beta = \frac{\delta-\alpha-\nu}{\gamma}$ with $\frac{|F|-1}{|F|}(1-(1-\hat{\beta})^3-\frac{27|F|}{m})$ evaluated at $\hat{\beta} = \delta-\alpha-\nu = 1-(\frac{|F|\epsilon}{|F|-1})^{1/3}+(\frac{27|F|}{n})^{1/3}$:

$$\frac{|F|-1}{|F|}\Big(1-(1-(1-(\frac{|F|\epsilon}{|F|-1})^{1/3}+(\frac{27|F|}{n})^{1/3}))^3-\frac{27|F|}{m}\Big)$$

Because of Fact 6.3.3, this expression is lower bounded by

$$\geq \frac{|F|-1}{|F|}\Big(1-(1-(1-(\frac{|F|\epsilon}{|F|-1})^{1/3}))^3+\frac{27|F|}{n}-\frac{27|F|}{m}\Big)$$

The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Therefore, this expression is more than

$$\begin{aligned}
&> \frac{|F|-1}{|F|}\Big(1-(1-(1-(\frac{|F|\epsilon}{|F|-1})^{1/3}))^3\Big)\\
&= \frac{|F|-1}{|F|}\Big(1-((\frac{|F|\epsilon}{|F|-1})^{1/3})^3\Big)\\
&= \frac{|F|-1}{|F|}\Big(1-\frac{|F|\epsilon}{|F|-1}\Big)\\
&= \frac{|F|-1}{|F|}-\epsilon
\end{aligned}$$

(It is apparent now that the choice of $\alpha$ at the start of the proof was made so that $\xi = \delta-\alpha-\nu-(\frac{27|F|}{n})^{1/3}$ would be the solution to $\frac{|F|-1}{|F|}(1-(1-\xi)^3)$

$= \frac{|F|-1}{|F|} - \epsilon$.)

When $\beta = 1$, $\frac{|F|-1}{|F|}(1 - (1-\beta)^3 - \frac{27|F|}{m}) > \frac{|F|-1}{|F|} - \frac{27|F|}{m} > \frac{|F|-1}{|F|} - \epsilon$ for large enough $n$ (again, note that $m > n$).

So, we have shown there is a situation in which the error is more than $\frac{|F|-1}{|F|} - \epsilon$. But this contradicts the fact that the recovery algorithm achieves $(3, \delta, \epsilon)$ on $\mathbf{C}$. ■

As an addendum, here are the small facts used above.

**Fact 6.3.3** *Let* $Z(\beta) \triangleq 1 - (1-\beta)^3$. *For any* $0 \leq \beta \leq 1$ *and* $0 \leq \rho \leq 1$ *such that* $1 - \beta - \rho \geq 0$, $Z(\beta + \rho) \geq Z(\beta) + \rho^3$.

**Proof:**

$$\begin{aligned}
&\Big(1-(1-(\beta+\rho))^3\Big)-\Big(1-(1-\beta)^3\Big)\\
&=(1-\beta)^3-\Big(1-(\beta+\rho)\Big)^3\\
&=\Big(1-3\beta+3\beta^2-\beta^3\Big)-\Big(1-3(\beta+\rho)+3(\beta+\rho)^2-(\beta+\rho)^3\Big)\\
&=3\rho+3\beta^2-\beta^3-3\Big(\beta^2+2\beta\rho+\rho^2\Big)+\Big(\beta^3+3\beta^2\rho+3\beta\rho^2+\rho^3\Big)\\
&=3\rho-6\beta\rho-3\rho^2+3\beta^2\rho+3\beta\rho^2+\rho^3\\
&=3\rho(1-2\beta-\rho+\beta^2+\beta\rho)+\rho^3\\
&=3\rho\Big((1-\beta)^2-\rho(1-\beta)\Big)+\rho^3\\
&=3\rho(1-\beta)(1-\beta-\rho)+\rho^3\\
&\geq\rho^3
\end{aligned}$$

■

**Fact 6.3.4** *Let* $Z(\beta)\triangleq 1-(1-\beta)^3$. *For any* $\beta\geq 0$ *and* $0\leq\rho\leq 1$ *with* $1-\beta-\rho\leq 1$, $Z(\beta+\rho)\leq Z(\beta)+4\rho$.

**Proof:** We can use the same first several steps of the Fact 6.3.3 to get

$$Z(\beta+\rho)-Z(\beta)=3\rho(1-\beta)(1-\beta-\rho)+\rho^3\leq 3\rho+\rho^3\leq 4\rho$$

■

By improving the adversary, we can get an even stronger bound for the non-binary case.

**Theorem 6.3.5** *Let* $\mathbf{C}$*:* $F^n \to F^m$ *be a linear* $(q = 3, \delta, \epsilon)$*-LDC with constant* $\epsilon > 0$*,* $\delta \leq \frac{|F|-1}{|F|}$*, and* $n$ *large enough. Then,* $m \geq 2^{.45\alpha n - 1}$ *where* $\alpha \triangleq \delta + \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3} + \frac{1}{|F|} - 1 - (\frac{108|F|}{n})^{1/3} - \nu$ *where* $\nu \triangleq \frac{10}{n}$*.*

**Note:** $\alpha > 0$ when $\frac{1}{|F|} + \epsilon > 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$ with $\phi(n) = (\frac{108|F|}{n})^{1/3} + \nu$. When $\frac{1}{|F|} + \epsilon > \mu + 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$ for some $\mu \geq 0$, then Fact 6.3.7 implies $\frac{1}{|F|} + \epsilon > 1 - 3\hat{\delta}(1-\hat{\delta})^2 - (1 - \frac{1}{|F|-1})3\hat{\delta}^2(1-\hat{\delta}) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\hat{\delta}^3 + \phi(n)$ for $\hat{\delta} = \delta - \frac{\mu}{4}$. Therefore, $\alpha > \frac{\mu}{4}$.

**Proof:** Claim 2.2 says that, without loss of generality, we can assume $\mathbf{C}$ has no codeword position that is identically zero.

For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \exists c \in F^* : \mathbf{C}(x)_j = cx_i \right\}$$

Also, for each $i \in [n]$, let $M_i$ be a largest matching of edges $\{j_1, j_2\} \subset [m]$ such that there exists a non-zero linear combination of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$ equalling $x_i$. For emphasis, no two edges in $M_i$ can intersect because it is a matching. Define $\alpha \triangleq \delta + \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3} + \frac{1}{|F|} - 1 - (\frac{108|F|}{n})^{1/3} - \nu$ where $\nu \triangleq \frac{10}{n}$. We will see the rationale for this choice at the end of the proof. If $\alpha \leq 0$: because $\epsilon > 0$ requires $m \geq 1$, we are done. So assume $\alpha > 0$. Now

consider:

$$S_1 \triangleq \left\{ i \in [n] \mid |R_i| \geq \nu m \right\}$$
$$S_2 \triangleq \left\{ i \in [n] \mid |M_i| \geq \frac{\alpha}{2} m \right\}$$

If $|S_2| \geq .9n$, then we can use Theorem 6.3.1 to conclude $m \geq 2^{.9*.5\alpha n - 1}$. If $|S_2| < .9n$, then because we know that $|S_1| \leq \frac{m}{\nu m} = .1n$, $\bar{S_1} \cap \bar{S_2}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S_1} \cap \bar{S_2}$. That is, $|R_1| < \nu m$ and $|M_1| < \frac{\alpha}{2} m$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Construct $\hat{M_1} \subset [m]$ as the union of all the edges in $M_1$. Note that $|\hat{M_1}| = 2|M_1|$.

Define $\gamma \triangleq \frac{|[m] \setminus (R_1 \cup \hat{M_1})|}{m}$ and $\beta \triangleq \min(\frac{\delta - \alpha - \nu}{\gamma}, 1)$. Let $A$ be a $(q, \delta, \epsilon)$ algorithm for $\mathbf{C}$. Let us consider the probability of error of the decoder over uniformly random $x \in F^n$ and the following distribution for the adversary. The adversary will choose $B_1 \subset [m] \setminus (R_1 \cup \hat{M_1})$ uniformly at random such that $|B_1| = \beta\gamma m$. The adversary will corrupt the codeword $\mathbf{C}(x)$ by adding an independent, uniformly random member of $F$ in the positions present in $R_1 \cup \hat{M_1}$ and adding an independent, uniformly random member of $F^*$ in the positions present in $B_1$. For simplicity, let $B \triangleq B_1 \cup R_1 \cup \hat{M_1}$, and let $\mathbf{C}(x) + B$ denote this corrupted codeword. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never

queries $R_1 \cup \hat{M}_1$.

Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1]$$
$$= \sum_{Q \subset [m], |Q| \leq 3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$. Define $Err_Q \triangleq$
$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound $Err_Q$ depending on all the different possibilities for $Q$ for which $\Pr[A \text{ queries } Q] > 0$.

- $x_1 \notin \text{span}(Q)$: By Theorem 5.1.1, $Err_Q \geq \frac{1}{|F|}$.
- $|Q| \leq 2$: $e_1 \in \text{span}(Q)$ implies that at least one bit in $Q$ is in $R_1 \cup \hat{M}_1$. But we have already said that $A$ never queries $R_1 \cup \hat{M}_1$.
- $|Q| = 3$ and $e_1 \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$: $e_1$ is not in the span of any two of the bits taken by themselves, because otherwise, at least one bit would be in $R_1 \cup \hat{M}_1$, and that would violate our assumption on $A$. Thus, there exists a non-zero linear combination of the three bits that equals $x_1$. Let us call it $L(\mathbf{C}(x)_{j_1}, \mathbf{C}(x)_{j_2}, \mathbf{C}(x)_{j_3}) = x_1$.

  Use the notation $B \cdot Q$ to mean $L\Big((\mathbf{C}(0) + B)_{j_1}, (\mathbf{C}(0) + B)_{j_2}, (\mathbf{C}(0) +$

$B)_{j_3}\Big)$. We can decompose $Err_Q$ into

$$Err_Q = \sum_{k \in F} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k] \cdot \Pr_B[B \cdot Q = k \mid A \text{ queries } Q]$$

For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k]$. We can further decompose $Err_{Q,k}$ into

$$Err_{Q,k} = \sum_{a,b,c \in F} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k; (\mathbf{C}(x)+B)_Q = abc] \cdot \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; B \cdot Q = k]$$

For simplicity, let us define:

$$q_{abc}^{Q,k} \triangleq \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; B \cdot Q = k]$$

Note that $L(a,b,c) = x_1 + k$. So we decompose even further. For fixed $a, b, c$,

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; B \cdot Q = k; (\mathbf{C}(x)+B)_Q = abc] = \sum_{r \in F : L(a,b,c) \neq r+k} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; B \cdot Q = k; (\mathbf{C}(x)+B)_Q = abc]$$

The event $B \cdot Q = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$= \sum_{r \in F : L(a,b,c) \neq r+k} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]$$

This motivates us to make the following definition. For $a, b, c \in F$,

$$p^Q_{abc}(r) \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abc]$$

Where $\sum_{r \in F} p^Q_{abc}(r) = 1$. This means,

$$Err_{Q,k} = \sum_{a,b,c \in F} \sum_{r \in F : L(a,b,c) \neq r+k} p^Q_{abc}(r) q^{Q,k}_{abc}$$

No two bits are equal, because otherwise, either one of the three bits would be in $\hat{M}_1$ or the third bit would be in $R_1$, which would also violate our assumption on $A$. Since there exists a non-zero linear combination of the three bits that equals $e_1$, the three bits are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x) + B)_{j_1}$, $(\mathbf{C}(x)+B)_{j_2}$, and $(\mathbf{C}(x)+B)_{j_3}$ are three independent, uniformly random bits. Thus, $\forall k, a, b, c$: $q^{Q,k}_{abc} = \frac{1}{|F|^3}$. So

$$\begin{aligned} Err_{Q,k} &= \sum_{a,b,c \in F} \sum_{r \in F : L(a,b,c) \neq r+k} p^Q_{abc}(r) \frac{1}{|F|^3} \\ &= \sum_{a,b,c \in F} (1 - p^Q_{abc}(L(a,b,c) - k)) \frac{1}{|F|^3} \end{aligned}$$

For simplicity, define $P_Q(k) = \sum_{a,b,c \in F} p^Q_{abc}(L(a,b,c) - k) \frac{1}{|F|^3}$. So $Err_{Q,k}$

$= 1 - P_Q(k)$. Therefore,

$$
\begin{aligned}
Err_Q &= \sum_{k \in F} (1 - P_Q(k)) \Pr_B[B \cdot Q = k \mid A \text{ queries } Q] \\
&\geq (|F| - 1) \min_{k \in F} \Pr_B[B \cdot Q = k \mid A \text{ queries } Q] \qquad (6.1) \\
&\qquad\qquad \text{because } \sum_{k \in F} P_Q(k) = 1 \\
&= (|F| - 1) \min_{k \in F} \sum_{l=0}^{3} \Pr_B[B \cdot Q = k \mid |B \cap Q| = l; A \text{ queries } Q] \cdot \\
&\qquad\qquad \Pr_B[|B \cap Q| = l \mid A \text{ queries } Q] \\
&= (|F| - 1) \min_{k \in F} \sum_{l=0}^{3} \Pr_B[B \cdot Q = k \mid |B \cap Q| = l; A \text{ queries } Q] \cdot \\
&\qquad\qquad \Pr_B[|B \cap Q| = l]
\end{aligned}
$$

Here is a table of values for $\Pr_B[B \cdot Q = k \mid |B \cap Q| = l; A \text{ queries } Q]$ depending on $k$ and $l$:

| | |
|---|---|
| $k = 0, l = 0 : 1$ | $k \neq 0, l = 0 : 0$ |
| $k = 0, l = 1 : 0$ | $k \neq 0, l = 1 : \frac{1}{\lvert F \rvert - 1}$ |
| $k = 0, l = 2 : \frac{1}{\lvert F \rvert - 1}$ | $k \neq 0, l = 2 : \frac{1}{\lvert F \rvert - 1}(1 - \frac{1}{\lvert F \rvert - 1})$ |
| $k = 0, l = 3 : \frac{1}{\lvert F \rvert - 1}(1 - \frac{1}{\lvert F \rvert - 1})$ | $k \neq 0, l = 3 : \frac{1}{\lvert F \rvert - 1}(1 - \frac{1}{\lvert F \rvert - 1}(1 - \frac{1}{\lvert F \rvert - 1}))$ |

Plugging these values in,

$$Err_Q \geq (|F|-1)\min_{k\in F}\sum_{l=0}^{3}\Pr_B[B\cdot Q=k \mid |B\cap Q|=l; A \text{ queries } Q]\cdot \Pr_B[|B\cap Q|=l]$$
$$= (|F|-1)\Big(\min(W_1,W_2)-3\frac{9}{\gamma m}\Big)$$

where $W_1$ and $W_2$ are defined as:

$$W_1 \triangleq (1-\beta)^3+\frac{1}{|F|-1}3\beta^2(1-\beta)+(\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2})\beta^3$$
$$W_2 \triangleq \frac{1}{|F|-1}3\beta(1-\beta^2)+(\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2})3\beta^2(1-\beta)+(\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2}+\frac{1}{(|F|-1)^3})\beta^3$$

Thus,

$$Err_Q \geq (|F|-1)\Big(\min(W_1-W_2,0)+W_2-\frac{27}{\gamma m}\Big)$$
$$= (|F|-1)\Big(\min\Big((1-\beta-\frac{\beta}{|F|-1})^3,0\Big)+\frac{1}{|F|-1}3\beta(1-\beta^2)+(\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2})3\beta^2(1-\beta)+(\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2}+\frac{1}{(|F|-1)^3})\beta^3-\frac{27}{\gamma m}\Big)$$

Since $\beta \leq \delta \leq \frac{|F|-1}{|F|}$, we have

$$\begin{aligned} Err_Q &\geq (|F|-1)\Big(\frac{1}{|F|-1}3\beta(1-\beta^2)+ \\ &\qquad (\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2})3\beta^2(1-\beta)+ \\ &\qquad (\frac{1}{|F|-1}-\frac{1}{(|F|-1)^2}+\frac{1}{(|F|-1)^3})\beta^3-\frac{27}{(|F|-1)\gamma m}\Big) \\ &> 3\beta(1-\beta)^2+(1-\frac{1}{|F|-1})3\beta^2(1-\beta)+ \\ &\qquad (1-\frac{1}{|F|-1}+\frac{1}{(|F|-1)^2})\beta^3-\frac{27|F|}{m} \end{aligned}$$

We have used that $\gamma m > (1-\delta)m > \frac{m}{|F|}$ in the last line. For simplicity, define

$$Z(\beta) \triangleq 3\beta(1-\beta)^2+(1-\frac{1}{|F|-1})3\beta^2(1-\beta)+(1-\frac{1}{|F|-1}+\frac{1}{(|F|-1)^2})\beta^3$$

Since for all $Q$, $Err_Q \geq Z(\beta)-\frac{27|F|}{m}$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq Z(\beta)-\frac{27|F|}{m}$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq Z(\beta)-\frac{27|F|}{m}$, where the probability is only over the internal coin flips of $A$.

Remember that $\beta = \frac{\delta-\alpha-\nu}{\gamma} \leq \frac{|F|-1}{|F|}$. First note that $Z(\beta)$ is strictly increasing in $\beta$. So

$$\begin{aligned} &Z(\frac{\delta-\alpha-\nu}{\gamma})-\frac{27|F|}{m} \geq \min\Big(Z(\delta-\alpha-\nu), Z(\frac{|F|-1}{|F|})\Big)-\frac{27|F|}{m} \\ &= \min\Big(Z(1-\frac{1}{|F|}-\epsilon^{1/3}(1-\frac{1}{|F|})^{2/3}+(\frac{108|F|}{n})^{1/3}), Z(\frac{|F|-1}{|F|})\Big)-\frac{27|F|}{m} \end{aligned}$$

For simplicity, define $\hat{\beta} = 1-\frac{1}{|F|}-\epsilon^{1/3}(1-\frac{1}{|F|})^{2/3}$. Because of Fact 6.3.6, $Z(\hat{\beta}+(\frac{108|F|}{n})^{1/3})-\frac{27|F|}{m}$ is lower bounded by $Z(\hat{\beta})+\frac{27|F|}{n}-\frac{27|F|}{m}$. The

lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Therefore,

$$\begin{aligned} & Z(\hat{\beta}) + \frac{27|F|}{n} - \frac{27|F|}{m} > Z(\hat{\beta}) \\ & = 3\hat{\beta}(1-\hat{\beta})^2 + (1 - \frac{1}{|F|-1})3\hat{\beta}^2(1-\hat{\beta}) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\hat{\beta}^3 \\ & = \frac{|F|-1}{|F|} - \epsilon \end{aligned}$$

The last line can be checked by algebra. (It is apparent now that the choice of $\alpha$ at the start of the proof was made so that $\xi = \delta - \alpha - \nu - (\frac{108|F|}{n})^{1/3}$ would be the solution to $3\xi(1-\xi)^2 + (1-\frac{1}{|F|-1})3\xi^2(1-\xi) + (1-\frac{1}{|F|-1}+\frac{1}{(|F|-1)^2})\xi^3 = \frac{|F|-1}{|F|} - \epsilon$.)

When $\beta = \frac{|F|-1}{|F|}$, $Z(\beta) - \frac{27|F|}{m} = \frac{|F|-1}{|F|} - \frac{27|F|}{m} > \frac{|F|-1}{|F|} - \epsilon$ for large enough $n$ (again, note that $m > n$).

So, we have shown there is a situation in which the error is more than $\frac{|F|-1}{|F|} - \epsilon$. But this contradicts the fact that the recovery algorithm achieves $(3, \delta, \epsilon)$ on $\mathbf{C}$. ■

As an addendum, here are the small facts used above.

**Fact 6.3.6** *Let* $Z(\beta) \triangleq 3\beta(1-\beta)^2 + (1 - \frac{1}{|F|-1})3\beta^2(1-\beta) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\beta^3$. *For any* $\beta$ *and* $\rho \geq 0$, $Z(\beta+\rho) \geq Z(\beta) + \frac{\rho^3}{4}$.

**Proof:** For easier notation, let $c \triangleq \frac{1}{|F|-1}$. We can rewrite $Z(\beta)$:

$$\begin{aligned}
Z(\beta) &= 3\beta(1-\beta)^2 + (1-c)3\beta^2(1-\beta) + (1-c+c^2)\beta^3 \\
&= 3\beta - 6\beta^2 + 3\beta^3 + (1-c)3\beta^2 - (1-c)3\beta^3 + (1-c+c^2)\beta^3 \\
&= 3\beta + (-3-3c)\beta^2 + (1+2c+c^2)\beta^3 \\
&= 3\beta - 3d\beta^2 + d^2\beta^3 \qquad \text{where } d \triangleq 1+c
\end{aligned}$$

Let $x \triangleq \beta + \rho/2$ and $p \triangleq \rho/2$. So now we need to lower bound $Z(x+p) - Z(x-p)$. Therefore,

$$\begin{aligned}
&Z(x+p) - Z(x-p) \\
&\quad = \Big(3(x+p) - 3d(x+p)^2 + d^2(x+p)^3\Big) - \\
&\qquad\qquad \Big(3(x-p) - 3d(x-p)^2 + d^2(x-p)^3\Big) \\
&\quad = 3\Big((x+p) - (x-p)\Big) - 3d\Big((x+p)^2 - (x-p)^2\Big) + \\
&\qquad\qquad d^2\Big((x+p)^3 - (x-p)^3\Big) \\
&\quad = 3(2p) - 3d(4xp) + d^2(6x^2p + 2p^3) \\
&\quad = 6p - 12dxp + 6d^2x^2p + 2d^2p^3 \\
&\quad = 3\rho - 6dx\rho + 3d^2x^2\rho + \frac{d^2}{4}\rho^3 \\
&\quad = 3\rho(1-dx)^2 + \frac{d^2}{4}\rho^3 \\
&\quad \geq \frac{\rho^3}{4}
\end{aligned}$$

■

**Fact 6.3.7** *Let* $Z(\beta) \triangleq 3\beta(1-\beta)^2 + (1-\frac{1}{|F|-1})3\beta^2(1-\beta) + (1-\frac{1}{|F|-1}+\frac{1}{(|F|-1)^2})\beta^3$. *For any* $\beta$ *and* $0 \leq \rho \leq 1$ *such that* $0 \leq \beta + \frac{\rho}{2} \leq \frac{2}{1+\frac{1}{|F|-1}}$, $Z(\beta+\rho) \leq Z(\beta) + 4\rho$.

**Proof:** We can use the same notation and the same first several steps of the Fact 6.3.6 to get

$$Z(x+p) - Z(x-p) = 3\rho(1-dx)^2 + \frac{d^2}{4}\rho^3 \leq 3\rho + \rho^3 \leq 4\rho$$

■

Remark: Since the length lower bounds for binary codes have depended on Theorem 6.1.1 and the length lower bounds for codes over arbitrary fields have depended on a generalization of Theorem 6.1.1, one might wonder if Theorem 6.1.1 can be improved. In fact it has been slightly improved, in Ben-Sasson and Viderman [9]. However, the improvement only exists for $\alpha = O(\frac{\log m}{m})$. The values of $\alpha$ in this range are too tiny to improve our proofs.

# Chapter 7

# Length Lower Bounds for Four Query, Linear, Binary LDCs

Now we extend our length lower bound techniques to four query, linear, binary LDCs. It is interesting that the bound we obtain for four query, linear, binary LDCs (length in terms of delta and epsilon) is very close to the bound we obtained earlier for three query, linear, binary LDCs.

**Theorem 7.1** *Let* $\mathbf{C}$*:* $\{0,1\}^n \rightarrow \{0,1\}^m$ *be a linear* $(q = 4, \delta, \epsilon)$*-LDC with constant* $\epsilon > 0$ *and* $n$ *large enough. Then,* $m \geq 2^{1.8\alpha n}$ *where* $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{96}{n})^{1/3} - \nu$ *where* $\nu \triangleq \frac{10}{n}$.

**Note:** $\alpha > 0$ when $\frac{1}{2} + \epsilon > 1 - 3\delta(1-\delta)^2 - \delta^3 + \phi(n)$ with $\phi(n) = (\frac{96}{n})^{1/3} + \nu$. When $\frac{1}{2} + \epsilon > \mu + 1 - 3\delta(1-\delta)^2 - \delta^3 + \phi(n)$ for some $\mu \geq 0$, then Fact 6.1.4 implies $\frac{1}{2} + \epsilon > 1 - 3(\delta - \frac{\mu}{4})(1 - (\delta - \frac{\mu}{4}))^2 - (\delta - \frac{\mu}{4})^3 + \phi(n)$. Therefore, $\alpha > \frac{\mu}{4}$.

**Proof:** Claim 2.2 says that, without loss of generality, we can assume $\mathbf{C}$ has no codeword position that is identically zero.

For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \mathbf{C}(x)_j = x_i \right\}$$

Also, for each $i \in [n]$, let $M_i$ be a largest matching of edges $\{j_1, j_2\} \subset [m]$ such that $\mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2} = x_i$. For emphasis, no two edges in $M_i$ can intersect because it is a matching. Let $\alpha \triangleq \delta + (\frac{\epsilon}{4})^{1/3} - \frac{1}{2} - (\frac{96}{n})^{1/3} - \nu$, where $\nu \triangleq \frac{10}{n}$. We will see the rationale for the choice of $\alpha$ at the end of the proof. If $\alpha \leq 0$: because $\epsilon > 0$ requires $m \geq 1$, we are done. So assume $\alpha > 0$. Now consider:

$$S_1 \triangleq \left\{ i \in [n] \mid |R_i| \geq \nu m \right\}$$
$$S_2 \triangleq \left\{ i \in [n] \mid |M_i| \geq \alpha m \right\}$$

If $|S_2| \geq .9n$, then we can use Theorem 6.1.1 to conclude $m \geq 2^{2*.9\alpha n}$. If $|S_2| < .9n$, then because we know that $|S_1| \leq \frac{m}{\nu m} = .1n$, $\bar{S_1} \cap \bar{S_2}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S_1} \cap \bar{S_2}$. That is, $|R_1| < \nu m$ and $|M_1| < \alpha m$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

We now construct a node cover $\hat{M_1} \subset [m]$ of those edges $\{j_1, j_2\} \subset [m]$ such that $\mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2} = x_i$. The graph formed by these edges a union of complete bipartite subgraphs. In each complete bipartite subgraph, the vertices on one of its sides all correspond to the same vector in $\{0,1\}^n$ – call it $a$. The vertices on the other side all correspond to $a + e_1$. Therefore, to construct

a small node cover, we can take the union of the vertices of the smaller side of each complete bipartite subgraph. Now, since $M_1$ is a largest matching of these edges, it has one member for each vertex on the smaller side of each complete bipartite subgraph, as well. Therefore, $|\hat{M}_1| = |M_1|$.

Define $\gamma \triangleq \frac{|[m]\setminus(R_1 \cup \hat{M}_1)|}{m}$ and $\beta \triangleq \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$. Let $A$ be a $(q, \delta, \epsilon)$ algorithm for $\mathbf{C}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus (R_1 \cup \hat{M}_1)$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1 \cup \hat{M}_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1 \cup \hat{M}_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. Make the following transformation on $A$ (which will let us combine cases in our analysis below): whenever $A$ queries fewer than four positions, have it arbitrarily query more so instead it queries four. This transformation cannot reduce the correctness of $A$, because $A$ can ignore the extra values it gets. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1 \cup \hat{M}_1$.

Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1]$$
$$= \sum_{Q \subset [m], |Q|=4} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$. Define $Err_Q \triangleq$
$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound $Err_Q$ depending on all the different possibilities for $Q$ for which $\Pr[A \text{ queries } Q] > 0$. First we give some notation.

Write $Q = \{j_1, j_2, j_3, j_4\}$. For $Q$ and $a, b, c, d \in \{0, 1\}$ such that $\Pr_{x,B}[A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abcd] > 0$, define

$$p^Q_{abcd} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abcd]$$

For simplicity, let us define the following notation. For a given $S \subseteq Q$,

$$q^{Q,k}_{abcd}(S) \triangleq \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = abcd \mid A \text{ queries } Q; |B \cap S| = k]$$

On an intuitive level, the cases below have similarities to each other, but they use different $S$ in their analyses. Here are the possibilities for $Q$:

- $x_1 \notin \text{span}(Q)$: By Theorem 5.1.1, $Err_Q \geq \frac{1}{2}$.
- $e_1$ is in the span of three of the vectors representing the bits, say $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$, taken by themselves; and $a_{j_4} \notin \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$: Define

$\hat{Q} \triangleq \{j_1, j_2, j_3\}$. Then,

$$Err_Q = \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \hat{Q}| = k] \cdot \Pr_B[|B \cap \hat{Q}| = k \mid A \text{ queries } Q]$$

$$= \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \hat{Q}| = k] \cdot \Pr_B[|B \cap \hat{Q}| = k]$$

Note that for any $Q$ and $0 \leq k \leq 3$, the events $A$ queries $Q$ and $|B \cap \hat{Q}| = k$ are independent. So for any $Q$ and $0 \leq k \leq 3$, $\Pr[A \text{ queries } Q; |B \cap Q| = \hat{k}] > 0$. Thus, above we are conditioning on events with nonzero probability. The second equality above also holds because of the independence of $A$ queries $Q$ and $|B \cap \hat{Q}| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \hat{Q}| = k]$. We can further decompose $Err_{Q,k}$ into

$$Err_{Q,k} = \sum_{a,b,c,d} q^{Q,k}_{abcd}(\hat{Q}) \cdot \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \hat{Q}| = k; (\mathbf{C}(x) + B)_Q = abcd]$$

$e_1$ is not in the span of any two of $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$ taken by themselves, because otherwise, at least one bit would be in $R_1 \cup \hat{M}_1$, and that would violate our assumption on $A$. Thus the sum of $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$ is $e_1$. (In this proof, additions involving codeword bits are implicitly modulo 2.)

So $a+b+c = x_1 + (k \bmod 2)$, and the above becomes:

$$\begin{aligned} Err_{Q,k} = & \sum_{\substack{a,b,c,d \\ a+b+c=k \bmod 2}} q^{Q,k}_{abcd}(\hat{Q}) \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap \hat{Q}| = k; (\mathbf{C}(x)+B)_Q = abcd] \\ & + \sum_{\substack{a,b,c,d \\ a+b+c=1+k \bmod 2}} q^{Q,k}_{abcd}(\hat{Q}) \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap \hat{Q}| = k; (\mathbf{C}(x)+B)_Q = abcd] \end{aligned}$$

The event $|B \cap \hat{Q}| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$\begin{aligned} Err_{Q,k} = & \sum_{\substack{a,b,c,d \\ a+b+c=k \bmod 2}} q^{Q,k}_{abcd}(\hat{Q}) \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd] \\ & + \sum_{\substack{a,b,c,d \\ a+b+c=1+k \bmod 2}} q^{Q,k}_{abcd}(\hat{Q}) \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd] \end{aligned}$$

This means,

$$Err_{Q,k} = \sum_{\substack{a,b,c,d \\ a+b+c=k \bmod 2}} (1 - p^Q_{abcd}) q^{Q,k}_{abcd}(\hat{Q}) + \sum_{\substack{a,b,c,d \\ a+b+c=1+k \bmod 2}} p^Q_{abcd} q^{Q,k}_{abcd}(\hat{Q})$$

No two bits of $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$ are equal, because otherwise, the third one would be in $R_1$, which would also violate our assumption on $A$. Since the sum of $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$ is $e_1$ and $a_{j_4} \notin \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$, $a_{j_1}$, $a_{j_2}$, $a_{j_3}$, and $a_{j_4}$ are linearly independent. Since, also, $x$ is uniformly

random, Fact 5.1.2 shows that $(\mathbf{C}(x)+B)_{j_1}$, $(\mathbf{C}(x)+B)_{j_2}$, $(\mathbf{C}(x)+B)_{j_3}$, and $(\mathbf{C}(x)+B)_{j_4}$ are four independent, uniformly random bits. Thus, $\forall k, a, b, c, d$: $q^{Q,k}_{abcd} = \frac{1}{16}$. So, when $k$ is even,

$$\begin{aligned} Err_{Q,k} = \Big((1-p^Q_{0000}) + (1-p^Q_{0110}) + (1-p^Q_{1100}) + (1-p^Q_{1010}) + \\ p^Q_{1000} + p^Q_{0100} + p^Q_{0010} + p^Q_{1110} + \\ (1-p^Q_{0001}) + (1-p^Q_{0111}) + (1-p^Q_{1101}) + (1-p^Q_{1011}) + \\ p^Q_{1001} + p^Q_{0101} + p^Q_{0011} + p^Q_{1111}\Big)/16 \end{aligned}$$

For simplicity, call this last expression $P_Q$. On the other hand, when $k$ is odd,

$$\begin{aligned} Err_{Q,k} &= \Big(p^Q_{0000} + p^Q_{0110} + p^Q_{1100} + p^Q_{1010} + \\ &\qquad (1-p^Q_{1000}) + (1-p^Q_{0100}) + (1-p^Q_{0010}) + (1-p^Q_{1110}) + \\ &\qquad p^Q_{0001} + p^Q_{0111} + p^Q_{1101} + p^Q_{1011} + \\ &\qquad (1-p^Q_{1001}) + (1-p^Q_{0101}) + (1-p^Q_{0011}) + (1-p^Q_{1111})\Big)/16 \\ &= 1 - P_Q \end{aligned}$$

By Lemma 5.2.2, we know the following two things:

$$\begin{aligned}
&\Pr_B[|B \cap \hat{Q}| = 0] + \Pr_B[|B \cap \hat{Q}| = 2] \\
&\qquad \geq (1-\beta)^3 - \frac{9}{\gamma m} + 3\beta^2(1-\beta) - \frac{9}{\gamma m} \\
&\qquad \geq (1-\beta)^3 + 3\beta^2(1-\beta) - \frac{36}{m} \\
&\Pr_B[|B \cap \hat{Q}| = 1] + \Pr_B[|B \cap \hat{Q}| = 3] \\
&\qquad \geq 3\beta(1-\beta)^2 - \frac{9}{\gamma m} + \beta^3 - \frac{9}{\gamma m} \\
&\qquad \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}
\end{aligned}$$

Combining everything, we find

$$\begin{aligned}
Err_Q &\geq \Big((1-\beta)^3 + 3\beta^2(1-\beta) - \frac{36}{m}\Big)P_Q + \\
&\qquad \Big(3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}\Big)(1-P_Q) \\
&= 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} + \\
&\qquad \Big((1-\beta)^3 + 3\beta^2(1-\beta) - 3\beta(1-\beta)^2 - \beta^3\Big)P_Q \\
&= 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} + \Big((1-\beta) - \beta\Big)^3 P_Q
\end{aligned}$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}$.

- $e_1$ is in the span of three of the vectors representing the bits, say $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$, taken by themselves; and $a_{j_4} \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$: Since no member of $Q$ is in $R_1 \cup \hat{M}_1$, $\mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2} + \mathbf{C}(x)_{j_3} = x_1$. $\mathbf{C}(x)_{j_4}$ cannot be the sum of the other three members of $Q$, because otherwise, $j_4$ would be in $R_1$, and that would violate our assumption on $A$. $\mathbf{C}(x)_{j_4}$

cannot be the sum of two other members of $Q$ – say $\mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2} = \mathbf{C}(x)_{j_4}$, because otherwise, $j_3$ or $j_4$ would be in $\hat{M}_1$, which would also violate our assumption on $A$. So $\mathbf{C}(x)_{j_4}$ equals one of the other members – assume it is $\mathbf{C}(x)_{j_1}$.

Define $Z$ as the event that either both $j_1$ and $j_4$ are corrupted or neither are. Define $\hat{Q} \triangleq \{j_1, j_2, j_3\}$. Consider the decomposition:

$$\begin{aligned}
Err_Q &= \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z} \mid A \text{ queries } Q] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z] \Pr_B[Z \mid A \text{ queries } Q] \\
&= \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z} \mid A \text{ queries } Q] + \\
&\quad \Big( \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k] \cdot \\
&\qquad \Pr_B[|B \cap \hat{Q}| = k \mid A \text{ queries } Q; Z] \Big) \Pr_B[Z \mid A \text{ queries } Q]
\end{aligned}$$

Note that for any $Q$, the events $A$ queries $Q$ and $Z$ are independent. So for any $Q$, $\Pr[A \text{ queries } Q; Z] > 0$. Thus, above we are conditioning on events with nonzero probability. For any $Q$ and $k$, $\Pr[|B \cap \hat{Q}| = k; A \text{ queries } Q; Z] > 0$, so we are not conditioning on zero probability events after the second equality, either. Because the events $Z$ and $A$

queries $Q$ are independent, the expression above becomes

$$Err_Q = \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z}] + \Big(\sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k] \cdot \Pr_B[|B \cap \hat{Q}| = k \mid A \text{ queries } Q; Z]\Big) \Pr_B[Z]$$

Let us first consider the error conditioned on $\bar{Z}$. For simplicity, define, $Err_{Q,\bar{Z}} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}]$. We can further decompose $Err_{Q,\bar{Z}}$ into

$$Err_{Q,\bar{Z}} = \sum_{a,b,c,d:d\neq a} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; (\mathbf{C}(x)+B)_Q = abcd] \cdot \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abcd \mid A \text{ queries } Q; \bar{Z}]$$

For simplicity, let us define (when $d \neq a$):

$$q_{abcd}^{Q,\bar{Z}} \triangleq \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abcd \mid A \text{ queries } Q; \bar{Z}]$$

So the above becomes:

$$
\begin{aligned}
&Err_{Q,\bar{Z}} = \\
&\sum_{a,b,c,d\neq a} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q;\bar{Z};(\mathbf{C}(x)+B)_Q = abcd] q_{abcd}^{Q,\bar{Z}} \\
&= \sum_{a,b,c,d\neq a} \Big( \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q;\bar{Z};j_1 \in B;(\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[j_1 \in B \mid A \text{ queries } Q;\bar{Z};(\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q;j_4 \in B;(\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[j_4 \in B \mid A \text{ queries } Q;\bar{Z};(\mathbf{C}(x)+B)_Q = abcd]\Big) q_{abcd}^{Q,\bar{Z}}
\end{aligned}
$$

If $A$ queries $Q$, $\bar{Z}$, and $(\mathbf{C}(x)+B)_Q = abcd$, over random $x$ and $B$, then it is equally likely that $j_1 \in B$ or $j_4 \in B$. So

$$
\begin{aligned}
&Err_{Q,\bar{Z}} = \sum_{a,b,c,d\neq a} \frac{1}{2}\Big( \\
&\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q;\bar{Z};j_1 \in B;(\mathbf{C}(x)+B)_Q = abcd] + \\
&\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q;\bar{Z};j_4 \in B;(\mathbf{C}(x)+B)_Q = abcd]\Big)
\end{aligned}
$$

Now, conditioning on whether $|\{j_2, j_3\} \cap B| \bmod 2$ is 0 or 1, and using

that $a + b + c = x_1 + |\{j_2, j_3\} \cap B| \bmod 2$:

$$
\begin{aligned}
Err_{Q,\bar{Z}} = \sum_{a,b,c,d \neq a} \frac{1}{2} \Big( & \\
& \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
& \qquad |\{j_2, j_3\} \cap B| \bmod 2 = 0; (\mathbf{C}(x)+B)_Q = abcd] \cdot \\
& \quad \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
& \qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
& \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
& \qquad |\{j_2, j_3\} \cap B| \bmod 2 = 1; (\mathbf{C}(x)+B)_Q = abcd] \cdot \\
& \quad \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
& \qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
& \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
& \qquad |\{j_2, j_3\} \cap B| \bmod 2 = 0; (\mathbf{C}(x)+B)_Q = abcd] \cdot \\
& \quad \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
& \qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
& \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
& \qquad |\{j_2, j_3\} \cap B| \bmod 2 = 1; (\mathbf{C}(x)+B)_Q = abcd] \cdot \\
& \quad \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
& \qquad (\mathbf{C}(x)+B)_Q = abcd] \Big) q_{abcd}^{Q,\bar{Z}}
\end{aligned}
$$

The events $\bar{Z}$, $j_1 \in B$, $j_4 \in B$, and $|\{j_2, j_3\} \cap B| \bmod 2 = 0$ are each independent of the internal randomness of $A$. Therefore, by Lemma

5.2.4,

$$
\begin{aligned}
&Err_{Q,\bar{Z}} = \\
&\sum_{a,b,c,d\neq a} \frac{1}{2}\Big( \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd]\Big) q^{Q,\bar{Z}}_{abcd}
\end{aligned}
$$

We notice that

$$\begin{aligned}
&\Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x) + B)_Q = abcd] \\
&= \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 1 \mid \bar{Z}; j_1 \in B; (\mathbf{C}(x) + B)_Q = abcd] \\
&= \Pr_{x,B}[x_1 = a + b + c \mid \bar{Z}; j_1 \in B; (\mathbf{C}(x) + B)_Q = abcd] \\
&= \Pr_{x,B}[x_1 = a + b + c \mid \bar{Z}; j_4 \in B; (\mathbf{C}(x) + B)_Q = dbca] \\
&= \Pr_{x,B}[x_1 = d + b + c \mid \bar{Z}; j_4 \in B; (\mathbf{C}(x) + B)_Q = abcd] \\
&= \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 1 \mid \bar{Z}; j_4 \in B; (\mathbf{C}(x) + B)_Q = abcd] \\
&= \Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
&\qquad\qquad (\mathbf{C}(x) + B)_Q = abcd]
\end{aligned}$$

Likewise, it is also true that

$$\begin{aligned}
&\Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x) + B)_Q = abcd] = \\
&\Pr_{x,B}[|\{j_2, j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_4 \in B; \\
&\qquad\qquad (\mathbf{C}(x) + B)_Q = abcd]
\end{aligned}$$

This gives

$$
\begin{aligned}
&Err_{Q,\bar{Z}} \\
&= \sum_{a,b,c,d\neq a} \frac{1}{2}\Big( \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\}\cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\}\cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\}\cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\cdot \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\}\cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd]\Big) q^{Q,\bar{Z}}_{abcd}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{a,b,c,d\neq a} \frac{1}{2}\Big( \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b+c \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]\Big) \cdot \\
&\quad \Big(\Pr_{x,B}[|\{j_2,j_3\} \cap B| \bmod 2 = 0 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd] + \\
&\qquad \Pr_{x,B}[|\{j_2,j_3\} \cap B| \bmod 2 = 1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abcd]\Big) q^{Q,\bar{Z}}_{abcd} \\
&= \sum_{a,b,c,d\neq a} \frac{1}{2} q^{Q,\bar{Z}}_{abcd} \\
&= \frac{1}{2}
\end{aligned}
$$

Also, Lemma 5.2.2 says $\Pr_B[\bar{Z}]$ is at least $2\beta(1-\beta) - \frac{4}{\gamma m} \geq \beta(1-\beta) - \frac{8}{m}$.

Let us now consider the error conditioned on $Z$: For simplicity, define, $Err_{Q,Z,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k]$. We can further decompose $Err_{Q,Z,k}$ into

$$
\begin{aligned}
Err_{Q,Z,k} = \sum_{a,b,c,d:d=a} &\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k; \\
&\qquad (\mathbf{C}(x)+B)_Q = abcd] \cdot \\
&\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abcd \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k]
\end{aligned}
$$

For simplicity, let us define (when $d = a$):

$$q_{abcd}^{Q,Z,k} \triangleq \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = abcd \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k]$$

Note that $a + b + c = x_1 + (k \bmod 2)$. So the above becomes:

$$\begin{aligned} Err_{Q,Z,k} = \sum_{\substack{a,b,c,d=a \\ a+b+c=k \bmod 2}} & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; Z; \\ & |B \cap \hat{Q}| = k; (\mathbf{C}(x) + B)_Q = abcd] q_{abcd}^{Q,Z,k} + \\ \sum_{\substack{a,b,c,d=a \\ a+b+c=1+k \bmod 2}} & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; Z; \\ & |B \cap \hat{Q}| = k; (\mathbf{C}(x) + B)_Q = abcd] q_{abcd}^{Q,Z,k} \end{aligned}$$

The event $Z \cap |B \cap \hat{Q}| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$\begin{aligned} Err_{Q,Z,k} = & \sum_{\substack{a,b,c,d=a \\ a+b+c=k \bmod 2}} q_{abcd}^{Q,Z,k} \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abcd] + \\ & \sum_{\substack{a,b,c,d=a \\ a+b+c=1+k \bmod 2}} q_{abcd}^{Q,Z,k} \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abcd] \end{aligned}$$

This means,

$$Err_{Q,Z,k} = \sum_{\substack{a,b,c,d=a \\ a+b+c=k \bmod 2}} (1 - p_{abcd}^{Q}) q_{abcd}^{Q,Z,k} + \sum_{\substack{a,b,c,d \\ a+b+c=1+k \bmod 2}} p_{abcd}^{Q} q_{abcd}^{Q,Z,k}$$

The vectors $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$ are linearly independent since none of $j_1$, $j_2$, or $j_3$ are in $R_1 \cup \hat{M}_1$ and $e_1 \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x) + B)_{j_1}$, $(\mathbf{C}(x) + B)_{j_2}$, and $(\mathbf{C}(x) + B)_{j_3}$ are three independent, uniformly random bits. Thus, $\forall k, a, b, c, d = a$: $q^{Q,k}_{abcd} = \frac{1}{8}$. So, when $k$ is even,

$$Err_{Q,Z,k} = \Big((1 - p^Q_{0000}) + (1 - p^Q_{0110}) + (1 - p^Q_{1101}) + (1 - p^Q_{1011}) + p^Q_{1001} + p^Q_{0100} + p^Q_{0010} + p^Q_{1111}\Big)/8$$

For simplicity, call this last expression $P_Q$. On the other hand, when $k$ is odd,

$$\begin{aligned} Err_{Q,Z,k} &= \Big(p^Q_{0000} + p^Q_{0110} + p^Q_{1101} + p^Q_{1011} + (1 - p^Q_{1001}) + (1 - p^Q_{0100}) + (1 - p^Q_{0010}) + (1 - p^Q_{1111})\Big)/8 \\ &= 1 - P_Q \end{aligned}$$

By Lemma 5.2.2, the probability that no bits were corrupted is at least $(1 - \beta)^4 - \frac{16}{\gamma m}$. Fact 7.2 shows that the probability that exactly one of $j_1$, $j_2$, and $j_3$ was corrupted (with $j_1$ and $j_4$ being either both corrupted or both not corrupted) is at least $2\beta(1 - \beta)^3 + \beta^2(1 - \beta)^2 - \frac{7}{\gamma m}$. Fact 7.3 shows that the probability that exactly two of $j_1$, $j_2$, and $j_3$ was corrupted (with $j_1$ and $j_4$ being either both corrupted or both not corrupted) is at least $2\beta^2(1 - \beta)^2 + \beta^3(1 - \beta) - \frac{6}{\gamma m}$. By Lemma 5.2.2, the probability

that all four bits were corrupted is at least $\beta^4 - \frac{16}{\gamma m}$. Therefore,

$$\Big(\Pr_B[|B \cap \hat{Q}| = 0 \mid Z] + \Pr_B[|B \cap \hat{Q}| = 2 \mid Z]\Big) \Pr_B[Z] \geq (1-\beta)^4 + 2\beta^2(1-\beta)^2 + \beta^3(1-\beta) - \frac{44}{m}$$

$$\Big(\Pr_B[|B \cap \hat{Q}| = 1 \mid Z] + \Pr_B[|B \cap \hat{Q}| = 3 \mid Z]\Big) \Pr_B[Z] \geq \beta(1-\beta)^3 + \beta^2(1-\beta)^2 + \beta^4 - \frac{46}{m}$$

Combining everything, we find

$$\begin{aligned}
Err_Q &= \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z}] + \\
&\qquad \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \hat{Q}| = k] \cdot \\
&\qquad\qquad \Pr_B[|B \cap \hat{Q}| = k \mid Z] \Pr_B[Z] \\
&\geq \Big(2\beta(1-\beta) - \frac{8}{m}\Big)\frac{1}{2} + \Big((1-\beta)^4 + 2\beta^2(1-\beta)^2 + \\
&\qquad\qquad \beta^3(1-\beta) - \frac{44}{m}\Big) P_Q + \\
&\qquad \Big(2\beta(1-\beta)^3 + \beta^2(1-\beta)^2 + \beta^4 - \frac{46}{m}\Big)(1 - P_Q) \\
&> \beta(1-\beta) + 2\beta(1-\beta)^3 + \beta^2(1-\beta)^2 + \beta^4 - \frac{50}{m} + \\
&\qquad \Big((1-\beta)^4 + 2\beta^2(1-\beta)^2 + \beta^3(1-\beta) - 2\beta(1-\beta)^3 - \\
&\qquad\qquad \beta^2(1-\beta)^2 - \beta^4\Big) P_Q
\end{aligned}$$

Because $\beta \leq \frac{1}{2}$,

$$\begin{aligned} Err_Q &\geq \beta(1-\beta) + 2\beta(1-\beta)^3 + \beta^2(1-\beta)^2 + \beta^4 - \frac{50}{m} \\ &= \beta - \beta^2 + 2\beta - 6\beta^2 + 6\beta^3 - 2\beta^4 + \beta^2 - 2\beta^3 + \beta^4 + \beta^4 - \frac{50}{m} \\ &= 3\beta - 6\beta^2 + 4\beta^3 - \frac{50}{m} \\ &\geq 3\beta(1-\beta)^2 + \beta^3 - \frac{50}{m} \end{aligned}$$

- $|Q| = 4$ and $e_1 \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}, a_{j_4}\}$; but $e_1$ is not in the span of any three of those vectors taken by themselves: We can decompose $Err_Q$ into

$$\begin{aligned} Err_Q = \sum_{k=0}^{4} &\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k] \cdot \\ &\Pr_B[|B \cap Q| = k \mid A \text{ queries } Q] \\ = \sum_{k=0}^{4} &\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k] \cdot \\ &\Pr_B[|B \cap Q| = k] \end{aligned}$$

Note that for any $Q$ and $0 \leq k \leq 4$, the events $A$ queries $Q$ and $|B \cap Q| = k$ are independent. So for any $Q$ and $0 \leq k \leq 4$, $\Pr[A \text{ queries } Q; |B \cap Q| = k] > 0$. Thus, above we are conditioning on events with nonzero probability. The second equality above also holds because of the independence of $A$ queries $Q$ and $|B \cap Q| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k]$. We can

further decompose $Err_{Q,k}$ into

$$Err_{Q,k} = \sum_{a,b,c,d} q^{Q,k}_{abcd}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abcd]$$

The sum of the four bits is $x_1$. So $a+b+c+d = x_1 + (k \bmod 2)$, and the above becomes:

$$Err_{Q,k} = \sum_{\substack{a,b,c,d \\ a+b+c+d=k \bmod 2}} q^{Q,k}_{abcd}(Q)$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abcd]$$
$$+ \sum_{\substack{a,b,c,d \\ a+b+c+d=1+k \bmod 2}} q^{Q,k}_{abcd}(Q)$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abcd]$$

The event $|B \cap Q| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$Err_{Q,k} = \sum_{\substack{a,b,c,d \\ a+b+c+d=k \bmod 2}} q^{Q,k}_{abcd}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd] +$$
$$\sum_{\substack{a,b,c,d \\ a+b+c+d=1+k \bmod 2}} q^{Q,k}_{abcd}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abcd]$$

This means,

$$Err_{Q,k} = \sum_{\substack{a,b,c,d \\ a+b+c+d=k \bmod 2}} (1 - p^Q_{abcd}) q^{Q,k}_{abcd}(Q) + \sum_{\substack{a,b,c,d \\ a+b+c+d=1+k \bmod 2}} p^Q_{abcd} q^{Q,k}_{abcd}(Q)$$

No two bits are equal, because otherwise, either the third or the fourth one would be in $\hat{M}_1$, and that would violate our assumption on $A$. No set of three bits sum to zero, because otherwise, the fourth one would be in $R_1$, which would also violate our assumption on $A$. And since the sum of the four bits is $e_1$, the bits are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x)+B)_{j_1}$, $(\mathbf{C}(x)+B)_{j_2}$, $(\mathbf{C}(x)+B)_{j_3}$, and $(\mathbf{C}(x)+B)_{j_4}$ are four independent, uniformly random bits. We will use this fact in each of the probability calculations below. Thus, $\forall k, a, b, c, d$: $q^{Q,k}_{abcd} = \frac{1}{16}$. So, when $k$ is even,

$$\begin{aligned} Err_{Q,k} = \Big( & p^Q_{0001} + p^Q_{0010} + p^Q_{0100} + p^Q_{1000} + \\ & (1 - p^Q_{1100}) + (1 - p^Q_{0110}) + (1 - p^Q_{0011}) + (1 - p^Q_{0000}) + \\ & p^Q_{1110} + p^Q_{1101} + p^Q_{1011} + p^Q_{0111} + \\ & (1 - p^Q_{0101}) + (1 - p^Q_{1001}) + (1 - p^Q_{1010}) + (1 - p^Q_{1111}) \Big) / 16 \end{aligned}$$

For simplicity, call this last expression $P_Q$. On the other hand, when $k$

is odd,

$$\begin{aligned} Err_{Q,k} = \Big((1-p^Q_{0001}) + (1-p^Q_{0010}) + (1-p^Q_{0100}) + (1-p^Q_{1000}) + \\ p^Q_{1100} + p^Q_{0110} + p^Q_{0011} + p^Q_{0000} + \\ (1-p^Q_{1110}) + (1-p^Q_{1101}) + (1-p^Q_{1011}) + (1-p^Q_{0111}) + \\ p^Q_{0101} + p^Q_{1001} + p^Q_{1010} + p^Q_{1111}\Big)/16 \\ = 1 - P_Q \end{aligned}$$

By Lemma 5.2.2, we know the following two things:

$$\begin{aligned} &\Pr_B[|B \cap Q| = 0] + \Pr_B[|B \cap Q| = 2] + \Pr_B[|B \cap Q| = 4] \\ &\qquad \geq (1-\beta)^4 - \frac{16}{\gamma m} + 6\beta^2(1-\beta)^2 - \frac{16}{\gamma m} + \beta^4 - \frac{16}{\gamma m} \\ &\Pr_B[|B \cap Q| = 1] + \Pr_B[|B \cap Q| = 3] \\ &\qquad \geq 4\beta(1-\beta)^3 - \frac{16}{\gamma m} + 4\beta^3(1-\beta) - \frac{16}{\gamma m} \end{aligned}$$

Combining everything, we find

$$\begin{aligned} Err_Q &\geq \Big((1-\beta)^4 + 6\beta^2(1-\beta)^2 + \beta^4 - \frac{96}{m}\Big)P_Q + \\ &\qquad \Big(4\beta(1-\beta)^3 + 4\beta^3(1-\beta) - \frac{64}{m}\Big)(1-P_Q) \\ &\geq \min\Big((1-\beta)^4 + 6\beta^2(1-\beta)^2 + \beta^4 - \frac{96}{m}, \\ &\qquad 4\beta(1-\beta)^3 + 4\beta^3(1-\beta) - \frac{64}{m}\Big) \\ &\geq \min\Big((1-\beta)^4 + 6\beta^2(1-\beta)^2 + \beta^4, \\ &\qquad 4\beta(1-\beta)^3 + 4\beta^3(1-\beta)\Big) - \frac{96}{m} \end{aligned}$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 4\beta(1-\beta)^3 + 4\beta^3(1-\beta) - \frac{96}{m} \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{96}{m}$.

Since for all $Q$, $Err_Q \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{96}{m}$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{96}{m}$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{96}{m}$, where the probability is only over the internal coin flips of $A$.

Remember that $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$. When $\beta = \frac{\delta-\alpha-\nu}{\gamma}$, first note that the expression $3\beta(1-\beta)^2 + \beta^3$ is strictly increasing in $\beta$. Therefore, we can lower bound $3\beta(1-\beta)^2+\beta^3-\frac{96}{m}$ evaluated at $\beta = \frac{\delta-\alpha-\nu}{\gamma}$ with $3\hat{\beta}(1-\hat{\beta})^2+\hat{\beta}^3-\frac{96}{m}$ evaluated at $\hat{\beta} = \delta - \alpha - \nu = \frac{1}{2} + (\frac{96}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}$:

$$3\Big(\frac{1}{2} + (\frac{96}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}\Big)\Big(\frac{1}{2} - (\frac{96}{n})^{1/3} + (\frac{\epsilon}{4})^{1/3}\Big)^2 + \Big(\frac{1}{2} + (\frac{96}{n})^{1/3} - (\frac{\epsilon}{4})^{1/3}\Big)^3 - \frac{96}{m}$$

Because of Fact 6.1.3, this expression is lower bounded by

$$\geq 3\Big(\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}\Big)\Big(\frac{1}{2} + (\frac{\epsilon}{4})^{1/3}\Big)^2 + \Big(\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}\Big)^3 + \frac{96}{n} - \frac{96}{m}$$
$$\geq \frac{1}{2} - \epsilon + \frac{96}{n} - \frac{96}{m}$$

The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Therefore, this expression is more than $\frac{1}{2} - \epsilon$. (It is apparent now that the choice of $\alpha$ at the start of the proof was made so that $\xi = \delta - \alpha - \nu - (\frac{96}{n})^{1/3}$ would be the solution to $3\xi(1-\xi)^2 + \xi^3 = \frac{1}{2} - \epsilon$.)

When $\beta = \frac{1}{2}$, $3\beta(1-\beta)^2 + \beta^3 - \frac{96}{m} = \frac{1}{2} - \frac{96}{m} > \frac{1}{2} - \epsilon$ for large enough $n$ (again, note that $m > n$).

So, we have shown there is a situation in which the error is more than $\frac{1}{2}-\epsilon$. But this contradicts the fact that the recovery algorithm achieves $(4, \delta, \epsilon)$ on $\mathbf{C}$. ■

Here are technical facts used in the above theorem.

**Fact 7.2** *For $k = 1$:*

$$\frac{\binom{2}{k}\binom{m-4}{\delta m-k} + k\binom{m-4}{\delta m-k-1}}{\binom{m}{\delta m}} > 2\delta(1-\delta)^3 + \delta^2(1-\delta)^2 - \frac{7}{m}$$

**Proof:** Using $k = 1$:

$$\begin{aligned}
\frac{2\binom{m-4}{\delta m-1} + \binom{m-4}{\delta m-2}}{\binom{m}{\delta m}} &= \frac{(\delta m)!(m-\delta m)!}{m!}(m-4)!\cdot \\
&\qquad \Big(\frac{2}{(\delta m-1)!(m-\delta m-3)!} + \frac{1}{(\delta m-2)!(m-\delta m-2)!}\Big) \\
&= \frac{1}{m(m-1)(m-2)(m-3)}\Big(2\delta m(m-\delta m)(m-\delta m-1)(m-\delta m-2)+ \\
&\qquad \delta m(\delta m-1)(m-\delta m)(m-\delta m-1)\Big) \\
&= \frac{\delta m(m-\delta m)}{m(m-1)(m-2)(m-3)}\cdot \\
&\qquad \Big(2(m-\delta m-1)(m-\delta m-2) + (\delta m-1)(m-\delta m-1)\Big) \\
&> \frac{\delta(1-\delta)}{m^2}\Big(2(m-\delta m-1)(m-\delta m-2) + (\delta m-1)(m-\delta m-1)\Big) \\
&> \frac{\delta(1-\delta)}{m^2}\Big(2(m-\delta m)^2 - 6(m-\delta m) + (\delta m)(m-\delta m) - \delta m - (m-\delta m)\Big) \\
&\geq \frac{\delta(1-\delta)}{m^2}\Big(2(m-\delta m)^2 + (\delta m)(m-\delta m) - 7m\Big) \\
&= \delta(1-\delta)\Big(2(1-\delta)^2 + \delta(1-\delta) - \frac{7}{m}\Big) \\
&\geq 2\delta(1-\delta)^3 + \delta^2(1-\delta)^2 - \frac{7}{m}
\end{aligned}$$

■

**Fact 7.3** *For $k = 2$:*

$$\frac{\binom{2}{k}\binom{m-4}{\delta m-k} + k\binom{m-4}{\delta m-k-1}}{\binom{m}{\delta m}} > \delta^2(1-\delta)^2 + 2\delta^3(1-\delta) - \frac{6}{m}$$

**Proof:** Using $k = 2$:

$$\begin{aligned}
\frac{\binom{m-4}{\delta m-2} + 2\binom{m-4}{\delta m-3}}{\binom{m}{\delta m}} &= \frac{(\delta m)!(m-\delta m)!}{m!}(m-4)!\cdot \\
&\qquad \Big(\frac{1}{(\delta m-2)!(m-\delta m-2)!} + \frac{2}{(\delta m-3)!(m-\delta m-1)!}\Big) \\
&= \frac{1}{m(m-1)(m-2)(m-3)}\Big(\delta m(\delta m-1)(m-\delta m)(m-\delta m-1) + \\
&\qquad 2\delta m(\delta m-1)(\delta m-2)(m-\delta m)\Big) \\
&= \frac{\delta m(\delta m-1)(m-\delta m)}{m(m-1)(m-2)(m-3)}\Big((m-\delta m-1) + 2(\delta m-2)\Big) \\
&= \frac{\delta m(\delta m-1)(m-\delta m)}{m(m-1)(m-2)(m-3)}\Big((m-\delta m) + 2(\delta m) - 5\Big) \\
&= \frac{\delta m(m-\delta m)}{m(m-1)(m-2)(m-3)}\cdot \\
&\qquad \Big(\delta m(m-\delta m) + 2(\delta m)^2 - 5\delta m - (m-\delta m) - 2(\delta m) + 5\Big) \\
&> \frac{\delta m(m-\delta m)}{m(m-1)(m-2)(m-3)}\Big(\delta m(m-\delta m) + 2(\delta m)^2 - 6m\Big) \\
&> \delta(1-\delta)\Big(\delta(1-\delta) + 2\delta^2 - \frac{6}{m}\Big) \\
&\geq \delta^2(1-\delta)^2 + 2\delta^3(1-\delta) - \frac{6}{m}
\end{aligned}$$

■

# Chapter 8

# Arbitrary Number of Queries for Special Classes of Decoders

In this chapter, we present extensions and adaptations of the methods of the previous chapter. We will make statements on the lengths of codes which have algorithms that can query any number of queries (potentially many more than three), but which must operate under certain commonly used restrictions. As in the previous chapter, we are requiring the algorithms we study to have high correctness. We can also make statements on the correctness of these codes, regardless of length.

**Theorem 8.1** *Let* $\mathbf{C} : F^n \to F^m$ *be a binary or linear code. Let* $A$ *be a linear decoder operating on* $\mathbf{C}$ *achieving correctness* $\frac{1}{|F|} + \epsilon \geq 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + O(\frac{1}{n^{1/3}})$ *for some* $\epsilon$ *and* $\delta$*. Then* $m \geq 2^{poly(n)}$*, where the exact form of the exponent depends on the value of* $3\delta(1-\delta)^2 + (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 - \frac{|F|-1}{|F|} + \epsilon - O(\frac{1}{n^{1/3}})$ *and whether we can assume* $\mathbf{C}$ *is binary or linear.*

**Proof:** This follows from using the three query length lower bounds: specifically either Theorem 6.2.2 if $\mathbf{C}$ is binary or Theorem 6.3.2 if $\mathbf{C}$ is linear. Consider the case in the proof of that respective theorem where $|Q| = 3$ and $e_1$ is

in the span of the vectors representing the positions in $Q$. The only additional property of $Q$ used was that the vectors representing the positions in $Q$ sum to $e_1$. We have assumed this in this theorem's statement. Therefore, the same error bound holds in this proof.

It remains to note that, for linear decoders operating on a codeword corrupted in the way specified in the three query length lower bound proofs, the error when $|Q| > 3$ is greater than or equal the error when $|Q| = 3$. ■

For a matching sum decoder, we can say more:

**Theorem 8.2** *Let* $\mathbf{C} : F^n \rightarrow F^m$ *be a binary or linear code. Let* $A$ *be a matching sum decoder operating on* $\mathbf{C}$ *achieving correctness* $\frac{1}{|F|} + \epsilon \geq 1 - 3\delta + O(\frac{1}{n^{1/3}})$ *for some* $\epsilon$ *and* $\delta$. *Then* $m \geq 2^{poly(n)}$, *where the exact form of the exponent depends on the value of* $3\delta - \frac{|F|-1}{|F|} + \epsilon - O(\frac{1}{n^{1/3}})$ *and whether we can assume* $\mathbf{C}$ *is binary and/or linear.*

**Proof:** Let $\alpha < \delta - \nu + \frac{\epsilon}{3} - \frac{|F|-1}{3|F|}$. As in the other length lower bound proofs, this comes into play both when utilizing the combinatorial lemma and in the final error calculation.

If $|S_2|$ is large, then we can use the respective combinatorial lemma (Theorem 6.1.1 if $\mathbf{C}$ is binary and linear, Theorem 6.2.1 if $\mathbf{C}$ is binary but not linear, or Theorem 6.3.1 if $\mathbf{C}$ is linear but not binary), to conclude $m \geq 2^{poly(n)}$.

Otherwise, $\bar{S}_1 \cap \bar{S}_2$ is nonempty. We construct a distribution for the adversary that corrupts the members of $R_1$ and $M_1$ in the same way as is done in the binary or linear length lower bound proofs. For $T \triangleq [m] \setminus (R_1 \cup M_1)$, however, the adversary behaves differently. It finds all the (non-intersecting) sets of $T$ which $A$ queries with nonzero probability. Because each such query set has size at least 3, and the decoder works with matchings, there are at most $m/3$ query sets. The adversary takes a node cover of these query sets and, uniformly at random, corrupts $(\delta - \alpha - \nu)m$ of these vertices. Thus, the algorithm errs with probability at least $\frac{(\delta-\alpha-\nu)m}{m/3} = 3(\delta - \alpha - \nu) > \frac{|F|-1}{|F|} - \epsilon$. ■

There are related correctness bounds to these length lower bounds for linear decoders and matching sum decoders. Please see the "Linear Decoders" section for more.

Here we introduce another definition that will lead to a reasonable restriction we can apply on algorithms.

**Definition 8.1** *For a linear code, an INDEPENDENT QUERY SET is a set of positions of the code such that the corresponding vectors are linearly independent.*

As part of our analysis on algorithms that query only independent query sets, first we will present a correctness bound on such algorithms and then we present a length lower bound on codes restricted to using them. The proofs of

both of these theorems are based on the proof of the three query linear codes length lower bound (Theorem 6.3.2).

**Claim 8.3** *Let* $\mathbf{C} : F^n \rightarrow F^m$ *be a linear code. Let* $A$ *be an algorithm operating on* $\mathbf{C}$ *which only queries independent query sets, such that the correct value of* $x_i$ *is always spanned by* $q$ *positions with non-zero coefficients. Then,* $\zeta_\delta(A) \leq 1 - q\delta + o(\delta) + O(\frac{1}{n})$.

**Proof:** Consider an adversary that chooses a set $B \subset [m]$ with $|B| = \delta m$ uniformly at random from all sets having that size. The adversary will corrupt $\mathbf{C}(x)$ by adding an independent, uniformly random member of $F^*$ in the positions present in $B$.

Consider the case where $|Q| = q$ and $e_1$ is in the span of the vectors representing the positions in $Q$. $e_1$ is not in the span of the vectors of any strict subset of $Q$, because the vectors corresponding to the positions in $Q$ are linearly independent. Following the steps in the three query linear length lower bound proof where we assume $q = 3$, we see by exactly the same reasoning that, in our case, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \geq 1 - q\delta + o(\delta) + O(\frac{1}{n})$.

It remains to note that, for algorithms that query only independent query sets operating on a codeword corrupted in the way specified above, the error when $|Q| > q$ is greater than or equal the error when $|Q| = q$. ■

**Theorem 8.4** *Let* $\mathbf{C} : F^n \rightarrow F^m$ *be a linear code. Let* $A$ *be an algorithm operating on* $\mathbf{C}$ *which only queries independent query sets. If, for a given* $\delta$ *fraction codeword corruption,* $A$ *achieves correctness* $\frac{1}{|F|}+\epsilon$*, then* $m \geq 2^{.45\alpha n-1}$ *where* $\alpha \triangleq \delta + \epsilon^{1/3}(1-\frac{1}{|F|})^{2/3} + \frac{1}{|F|} - 1 - (\frac{108|F|}{n})^{1/3} - \nu$*, where* $\nu \triangleq \frac{10}{n}$.

**Note:** $\alpha > 0$ when $\frac{1}{|F|} + \epsilon > 1 - 3\delta(1-\delta)^2 - (1-\frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$ with $\phi(n) = (\frac{108|F|}{n})^{1/3} + \nu$.

**Proof:** This follows from using the proof of the three query linear code length lower bound (Theorem 6.3.2). Most of that proof actually holds for all query sizes. In fact, the error bound (6.1) is stated for arbitrary independent query sets $Q$. The only part of the proof that requires $|Q| \leq 3$ is to guarantee that the query sets of size three are linearly independent. ■

# Chapter 9

# General Correctness Bounds

In this chapter, we present several correctness bounds that hold for algorithms with any number of queries. We appear to be among the first ones to investigate the trade off between the correctness of an LDC and the number of queries an LDC is allowed to make.

## 9.1 Linear Algebra Property

In this section, we prove a theorem that is key to improving the above correctness bound when a code is binary and linear. It in fact depends little on our particular set up and can be viewed as a linear algebra property.

Recall that we think of the $a_j$'s as column vectors of length $n$. So $(a_1 a_2 ... a_q)$ is an $n$ by $q$ matrix, and in what follows, $z \cdot (a_1 a_2 ... a_q)$ for $z \in \{0,1\}^n$ represents the multiplication of a length $n$ vector by a $n$ by $q$ matrix. Similarly, $(e_1 a_1 a_2 ... a_{r-1}) \cdot w$ for $w \in \{0,1\}^r$ represents the multiplication of an $n$ by $r$ matrix by a length $r$ vector.

**Theorem 9.1.1** *Let $a_1$, $a_2$, ... $a_q$ be non-zero vectors in $\{0,1\}^n$. Assume none of them are $e_1$ but, collectively, they span $e_1$. Then there exists a $v \in \{0,1\}^{n-1}$*

*such that the bit string* $(1v) \cdot (a_1 a_2 ... a_q)$ *has at most* $\frac{q}{2}$ *ones in it.*

**Proof:** Note that the conditions of the statement of the theorem imply that the rank of $a_1$, $a_2$, ... $a_q$, call it $r$, is at least 2. Since these vectors span $e_1$, we can, without loss of generality, relabel the indices on the $a$'s so that $\{e_1$, $a_1$, $a_2$, ... $a_{r-1}\}$ forms a basis. For $w \in \{0,1\}^r$, let

$$t_w \triangleq \left| \left\{ j \in [q] \mid a_j = (e_1 a_1 a_2 ... a_{r-1}) \cdot w \right\} \right|$$

Thus $t_w$ is the multiplicity with which $(e_1 a_1 a_2 ... a_{r-1}) \cdot w$ appears within $a_1$, $a_2$, ... $a_q$. By the conditions of the statement of the theorem, $t_{00...0} = t_{10...0} = 0$. Therefore

$$\sum_{\substack{w \in \{0,1\}^r \\ w \neq 00...0, 10...0}} t_w = q$$

For $b \in \{0,1\}^{r-1}$, define $N_b \triangleq \sum_{\substack{w \in \{0,1\}^r \\ <1b,w>=1, w \neq 00...0, 10...0}} t_w$.

The following claim will help us.

**Claim 9.1** *Fix* $b \in \{0,1\}^{r-1}$. *For all* $v \in \{0,1\}^{n-1}$ *such that* $(1v) \cdot (e_1 a_1 a_2 ... a_{r-1}) = 1b$, *the bit string* $(1v) \cdot (a_1 a_2 ... a_q)$ *has* $N_b$ *ones in it.*

**Proof:** Let $j \in [q]$ and $a_j = (e_1 a_1 a_2 ... a_{r-1}) \cdot w$ for a $w \in \{0,1\}^r$. This implies $< a_j, 1v > = < (e_1 a_1 a_2 ... a_{r-1}) \cdot w, 1v > = < (1v) \cdot (e_1 a_1 a_2 ... a_{r-1}), w > = < 1b, w >$. ■

To finish the proof of the theorem, let's consider:

$$\begin{aligned}
\sum_{b\in\{0,1\}^{r-1}} N_b &= \sum_{b\in\{0,1\}^{r-1}} \sum_{\substack{w\in\{0,1\}^r \\ <1b,w>=1, w\neq 00\ldots0,10\ldots0}} t_w \\
&= \sum_{\substack{w\in\{0,1\}^r \\ w\neq 00\ldots0,10\ldots0}} \sum_{\substack{b\in\{0,1\}^{r-1} \\ <1b,w>=1}} t_w \\
&= \sum_{\substack{w\in\{0,1\}^r \\ w\neq 00\ldots0,10\ldots0}} 2^{r-2} t_w \qquad \text{see below} \qquad (9.1) \\
&= 2^{r-2} q
\end{aligned}$$

Line (9.1) holds because, for any nonzero vector $z \in \{0,1\}^{r-1}$, exactly half of $b \in \{0,1\}^{r-1}$ have $b \cdot z = 1$.

We were summing over $2^{r-1}$ $N_b$'s. Therefore, there is at least one $b$ for which $N_b \leq \frac{2^{r-2}q}{2^{r-1}} = \frac{q}{2}$.

Because $e_1$, $a_1$, $a_2$, ... $a_{r-1}$ are linearly independent, for any vector $b' \in \{0,1\}^{r-1}$, there exists a vector $v \in \{0,1\}^{n-1}$ such that

$$(e_1 a_1 a_2 ... a_{r-1}) \cdot (1v) = 1b'$$

Therefore, there is a $v$ corresponding to $b$ such that $(a_1 a_2 ... a_q) \cdot (1v)$ has at most $\frac{q}{2}$ ones in it. ■

## 9.2 q Query, Binary, Linear LDCs

The following proof for binary, linear LDCs is similar to the proof of Theorem 9.5.1. But we get an improvement by utilizing Theorem 9.1.1.

**Theorem 9.2.1** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a linear code, and let* $A$ *be a* $q$ *query recovery algorithm for it. Then, for large enough* $n$*,*

$$\zeta_\delta(A) \leq 1 - 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 \rfloor - 1}\delta^{\lceil q/4 \rceil + 1}(1-\delta)^{\lfloor q/4 \rfloor} + \frac{5 * 2^{q-1}q^2}{n}$$

**Note:** We will actually prove the stronger, but more complicated, bound:

$$\zeta_\delta(A) \leq 1 - 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}\delta^{\lceil q/4 + 1/2 \rceil}(1-\delta)^{\lfloor q/4 - 1/2 \rfloor} + \frac{5 * 2^{q-1}q^2}{n}$$

**Proof:** By Claim 2.2, there exists a code **C'** with no codeword position identically zero and an algorithm $A'$ operating on it such that, for any $\delta$, $\zeta_\delta(A) \leq \zeta_\delta(A')$. In this proof, we will henceforth work exclusively with **C'** and $A'$, but, for simplicity, drop the primes from notation.

For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \mathbf{C}(x)_j = x_i \right\}$$

So there exists at least one $i$ such that $|R_i| \leq \frac{m}{n}$. Without loss of generality, assume $|R_1| \leq \frac{m}{n}$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Define $\gamma \triangleq \frac{|[m] \setminus R_1|}{m}$ and $\beta \triangleq \frac{\delta - \frac{|R_1|}{m}}{\gamma} = \frac{\delta - \frac{|R_1|}{m}}{1 - \frac{|R_1|}{m}}$. Note that if $\delta \geq \frac{1}{2}$, Claim 4.1 already gives the result. So we can assume $\delta \leq \frac{1}{2}$, and therefore $\beta \leq \frac{1}{2}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus R_1$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. We also note that, because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries $q$ positions. If the algorithm ever queried fewer than $q$ positions, have it query more and ignore the additional values obtained. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1$.

A property of $B$ that we will use later is that, for all query sets $Q$ of size $q$, value $r \in \{0,1\}$, and string $s \in \{0,1\}^q$, $\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r] > 0$. This is because $B$ is independent of $x$. Since all of these probabilities are positive, we will be able to condition on these events properly. Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$.

For $Q$ and $r \in \{0,1\}$ having $\Pr_x[A \text{ queries } Q; x_1 = r] > 0$, define $Err_{Q,r} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq r \mid A \text{ queries } Q; x_1 = r]$. Noting that the distributions of $x$ and the internal randomness of $A$ are independent, we decompose:

$$\begin{aligned}\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] &= \sum_Q (Err_{Q,0} \Pr_x[x_1 = 0] + Err_{Q,1} \Pr_x[x_1 = 1]) \cdot \\ &\qquad \Pr[A \text{ queries } Q] \\ &= \frac{1}{2} \sum_Q (Err_{Q,0} + Err_{Q,1}) \Pr[A \text{ queries } Q]\end{aligned}$$

So let us lower bound $Err_{Q,0} + Err_{Q,1}$:

$$\begin{aligned}&Err_{Q,0} + Err_{Q,1} \\ &= \sum_{s \in \{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; x_1 = 0; (\mathbf{C}(x)+B)_Q = s] \cdot \\ &\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0] \\ &\quad + \sum_{s \in \{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; x_1 = 1; (\mathbf{C}(x)+B)_Q = s] \cdot \\ &\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1]\end{aligned}$$

The value of $x_1$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4, we can remove the conditioning on the value of

$x_1$:

$$
\begin{aligned}
&Err_{Q,0} + Err_{Q,1} \\
&= \sum_{s\in\{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] \cdot \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0] \\
&\quad + \sum_{s\in\{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] \cdot \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1]
\end{aligned}
$$

For ease of notation, let us make the following definition.

$$
p_s^Q \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s]
$$

This gives

$$
\begin{aligned}
&Err_{Q,0} + Err_{Q,1} \\
&= \sum_{s\in\{0,1\}^q} (1 - p_s^Q) \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0] \\
&\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1]
\end{aligned}
$$

The internal randomness of $A$ is independent of $B$, and the values of the positions labeled by $Q$ are independent of whether the algorithm actually

queries $Q$. So we have

$$\begin{aligned}
Err_{Q,0} + Err_{Q,1} &= \sum_{s\in\{0,1\}^q} (1 - p_s^Q) \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \\
&\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] \\
&= \sum_{s\in\{0,1\}^q} \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \\
&\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Big( \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \Big) \\
&= 1 + \sum_{s\in\{0,1\}^q} p_s^Q \Big( \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \Big)
\end{aligned}$$

This expression is smallest when $p_s^Q$ is 0 for $s$ such that $\Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] > \Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]$ and $p_s^Q$ is 1 for $s$ such that $\Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] < \Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]$. Thus the last expression is lower bounded by

$$\begin{aligned}
&\geq 1 + \sum_{\substack{s\in\{0,1\}^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big( \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \Big)
\end{aligned}$$

where, to save space, we have used shorthand notation for specifying the last summation over $s$. Let us prove a side fact. It is clear that:

$$\sum_{s\in\Sigma^q} \Big( \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] \Big) = 1 - 1 = 0$$

This implies

$$\sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\Big) =$$
$$\sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=1]>\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big)$$

Therefore,

$$\sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\Big) =$$
$$\frac{1}{2}\sum_{s\in\Sigma^q} \Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big|$$

Plugging this in above,

$$Err_{Q,0} + Err_{Q,1} \geq$$
$$1 - \frac{1}{2}\sum_{s\in\{0,1\}^q} \Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big|$$

To simplify things, let us temporarily just operate on the sub-expression:

$$\begin{aligned}
\Delta &\triangleq \sum_{s\in\{0,1\}^q} \Big| \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big| \\
&= \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 1w] \Pr_x[x_2x_3...x_n = w \mid x_1 = 1] \\
&\qquad - \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w] \Pr_x[x_2x_3...x_n = w \mid x_1 = 0]\Big| \\
&= \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 1w] \Pr_x[x_2x_3...x_n = w] - \\
&\qquad\qquad \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w] \Pr_x[x_2x_3...x_n = w]\Big| \\
&= \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Pr_x[x_2x_3...x_n = w]\Big(\Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \\
&\qquad\qquad \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\Big)\Big| \\
&= \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Big(\Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\Big)\Big|
\end{aligned}$$

By Theorem 9.1.1, we can choose a $v \in \{0,1\}^n$ for which $\mathbf{C}(1v)_Q$ has weight at most $\frac{q}{2}$. Note that by the reductions we have made, no position in $Q$ always equals zero or always equals $x_1$. We will see the importance of this

choice of $v$ later. Using this $v$, we can rearrange the last line:

$$\Delta = \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Big( \Pr_B[B_Q = s - \mathbf{C}(1(w+v))_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q] \Big) \Big|$$

$$\leq \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \sum_{w\in\{0,1\}^{n-1}} \Big| \Pr_B[B_Q = s - \mathbf{C}(1(w+v))_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q] \Big) \Big|$$

$$= \frac{1}{2^{n-1}} \sum_{w\in\{0,1\}^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s - \mathbf{C}(1(w+v))_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q] \Big) \Big|$$

switch summations

Relabeling $s - \mathbf{C}(1(w+v))_Q$ by $s$, we have

$$= \frac{1}{2^{n-1}} \sum_{w\in\{0,1\}^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(1(w+v))_Q - \mathbf{C}(0w)_Q] \Big|$$

$$= \frac{1}{2^{n-1}} \sum_{w\in\{0,1\}^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(1v)_Q] \Big| \quad \mathbf{C} \text{ is linear}$$

We will bound

$$\bar{\Delta} \triangleq \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(1v)_Q] \Big|$$

Now will be the first time we use a fact about the distribution of $B$, excluding when we used that $B$ is independent of $A$. For a given $s$, $\Pr_B[B_Q = s]$ is a function of only the number of 1's in $s$. Let $\mathbf{C}(1v)_Q$ have $a \leq \frac{q}{2}$ 1's. Also,

let $b$ be the number of positions where $\mathbf{C}(1v)_Q$ and $s$ both equal 1; and let $c$ be number of positions where $\mathbf{C}(1v)_Q$ equals 0 but $s$ equals 1. Therefore, by Lemmas 5.2.2 and 5.2.3 (for large enough $m$), we have

$$\bar{\Delta} \leq \sum_{b=0}^{a} \binom{a}{b} \sum_{c=0}^{q-a} \binom{q-a}{c} (\left|\beta^{b+c}(1-\beta)^{q-b-c} - \beta^{a-b+c}(1-\beta)^{q-a+b-c}\right| + \frac{3q^2}{m})$$

Simplifying, we have

$$= \sum_{b=0}^{a} \binom{a}{b} \sum_{c=0}^{q-a} \binom{q-a}{c} \left|\beta^{b+c}(1-\beta)^{q-b-c} - \beta^{a-b+c}(1-\beta)^{q-a+b-c}\right| + \frac{3 * 2^q q^2}{m}$$

$$= \sum_{b=0}^{a} \binom{a}{b} \Big(\sum_{c=0}^{q-a} \binom{q-a}{c} \beta^c (1-\beta)^{q-a-c}\Big) \left|\beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b\right| + \frac{3 * 2^q q^2}{m}$$

$$= \sum_{b=0}^{a} \binom{a}{b} \left|\beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b\right| + \frac{3 * 2^q q^2}{m}$$

$\delta < \frac{1}{2}$ must hold for $A$ to have nontrivial correctness. Hence $\delta < 1 - \delta$ and $\beta < 1 - \beta$. Therefore, the expression inside of the absolute value is positive if and only if $b < \frac{a}{2}$. Also note that the value of the expression inside of the absolute value, for a given $b$, has the opposite sign of that same expression when $b$ is replaced by $a - b$. Using these facts, we have

$$\bar{\Delta} = 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \left(\beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b\right) + \frac{3 * 2^q q^2}{m}$$

Because $\sum_{b=0}^{a} \binom{a}{b}\beta^b(1-\beta)^{a-b} = (1-\beta+\beta)^a = 1$, the last line equals

$$= 2\Big(1 - \sum_{b=\lfloor (a+1)/2 \rfloor}^{a} \binom{a}{b}\beta^b(1-\beta)^{a-b} - \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b}\beta^{a-b}(1-\beta)^b\Big) + \frac{3 * 2^q q^2}{m}$$

$$\leq 2\Big(1 - 2\sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b}\beta^{a-b}(1-\beta)^b\Big) + \frac{3 * 2^q q^2}{m}$$

interchanging $b$ and $a-b$ in second term

$$< 2 - 4\binom{a}{\lfloor (a-1)/2 \rfloor}\beta^{\lceil (a+1)/2 \rceil}(1-\beta)^{\lfloor (a-1)/2 \rfloor} + \frac{3 * 2^q q^2}{m}$$

$$\leq 2 - 4\binom{\lfloor q/2 \rfloor}{\lfloor (\lfloor q/2 \rfloor - 1)/2 \rfloor}\beta^{\lceil (q/2+1)/2 \rceil}(1-\beta)^{\lfloor (q/2-1)/2 \rfloor} + \frac{3 * 2^q q^2}{m}$$

Note that for large $\delta$ and thus $\beta$, stronger upper bounds than the ones used in the last equation block could have been used instead.

Therefore, we have

$$
\begin{aligned}
&Err_{Q,0} + Err_{Q,1} \\
&\quad \geq 1 - \frac{1}{2}\Big(2 - 4\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}\beta^{\lceil q/4+1/2 \rceil}(1-\beta)^{\lfloor q/4-1/2 \rfloor} + \frac{3 * 2^q q^2}{m}\Big) \\
&\quad = 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}\beta^{\lceil q/4+1/2 \rceil}(1-\beta)^{\lfloor q/4-1/2 \rfloor} - \frac{3 * 2^{q-1} q^2}{m} \\
&\quad \geq 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}(\delta - \frac{1}{n})^{\lceil q/4+1/2 \rceil}(1-(\delta - \frac{1}{n}))^{\lfloor q/4-1/2 \rfloor} - \frac{3 * 2^{q-1} q^2}{m} \\
&\quad \geq 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}(\delta - \frac{1}{n})^{\lceil q/4+1/2 \rceil}(1-\delta)^{\lfloor q/4-1/2 \rfloor} - \frac{3 * 2^{q-1} q^2}{m} \\
&\quad \geq 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}(\delta^{\lceil q/4+1/2 \rceil} - \frac{q}{n})(1-\delta)^{\lfloor q/4-1/2 \rfloor} - \frac{3 * 2^{q-1} q^2}{m} \\
&\quad \geq 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}(\delta^{\lceil q/4+1/2 \rceil}(1-\delta)^{\lfloor q/4-1/2 \rfloor} - \frac{q}{n}) - \frac{3 * 2^{q-1} q^2}{m} \\
&\quad \geq 2\binom{\lfloor q/2 \rfloor}{\lfloor q/4 - 1/2 \rfloor}(\delta^{\lceil q/4+1/2 \rceil}(1-\delta)^{\lfloor q/4-1/2 \rfloor} - \frac{q}{n}) - \frac{3 * 2^{q-1} q^2}{m}
\end{aligned}
$$

The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Noting that $\frac{2\binom{\lfloor q/2 \rfloor}{\lfloor q/4-1/2 \rfloor}q}{n} \leq 2\frac{2^{\lfloor q/2 \rfloor}q}{n} \leq \frac{2*2^{q-1}q^2}{n}$ for $q > 1$, the result is obtained. ■

## 9.3 Probabilistic Method for the Non-Linear Case

The following theorem is a non-linear analog of the linear algebra property we proved earlier for linear codes (Theorem 9.1.1).

**Theorem 9.3.1** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a code, and let* $Q$ *be a size* $q$ *query set an algorithm* $A$ *queries. Also assume that for every position* $j \in [m]$*,*

$|\Pr_{x\in\{0,1\}^n}[x_1 = \mathbf{C}(x)_j] - \Pr_{x\in\{0,1\}^n}[x_1 \neq \mathbf{C}(x)_j]| \leq t$. *Then*

$$\mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq q(\frac{1}{2} + 3t)$$

**Proof:** To make things compact, define $a \triangleq \Pr_{w\in\{0,1\}^{n-1}}[\mathbf{C}(1w)_j = 1]$ and $b \triangleq \Pr_{w\in\{0,1\}^{n-1}}[\mathbf{C}(0w)_j = 0]$. Consider the following:

$$\begin{aligned}
&\Big|a + b - 1\Big| \\
&\quad = \Big|\frac{b}{2} + \frac{a}{2} - \frac{1-a}{2} - \frac{1-b}{2}\Big| \\
&\quad = \Big| \Pr_{x\in\{0,1\}^n}[\mathbf{C}(x)_j = 0 \mid x_1 = 0] \Pr_{x\in\{0,1\}^n}[x_1 = 0] + \\
&\qquad\qquad \Pr_{x\in\{0,1\}^n}[\mathbf{C}(x)_j = 1 \mid x_1 = 1] \Pr_{x\in\{0,1\}^n}[x_1 = 1] - \\
&\qquad\qquad \Pr_{x\in\{0,1\}^n}[\mathbf{C}(x)_j = 0 \mid x_1 = 1] \Pr_{x\in\{0,1\}^n}[x_1 = 1] - \\
&\qquad\qquad \Pr_{x\in\{0,1\}^n}[\mathbf{C}(x)_j = 1 \mid x_1 = 0] \Pr_{x\in\{0,1\}^n}[x_1 = 0]\Big| \\
&\quad = \Big| \Pr_{x\in\{0,1\}^n}[x_1 = \mathbf{C}(x)_j] - \Pr_{x\in\{0,1\}^n}[x_1 \neq \mathbf{C}(x)_j]\Big| \\
&\quad \leq t
\end{aligned}$$

We will use this fact shortly. Now consider:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \\
&\quad= \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} \sum_{j\in Q} d\Big(\mathbf{C}(1w_1)_j, \mathbf{C}(0w_2)_j\Big) \\
&\quad= \sum_{j\in Q} \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_j, \mathbf{C}(0w_2)_j\Big) \\
&\quad= \sum_{j\in Q} \Pr_{w_1,w_2\in\{0,1\}^{n-1}}[\mathbf{C}(1w_1)_j \neq \mathbf{C}(0w_2)_j] \\
&\quad= \sum_{j\in Q} \Big( \Pr_{w_1\in\{0,1\}^{n-1}}[\mathbf{C}(1w_1)_j = 1] \Pr_{w_2\in\{0,1\}^{n-1}}[\mathbf{C}(0w_2)_j = 0] + \\
&\qquad\qquad \Pr_{w_1\in\{0,1\}^{n-1}}[\mathbf{C}(1w_1)_j = 0] \Pr_{w_2\in\{0,1\}^{n-1}}[\mathbf{C}(0w_2)_j = 1] \Big)
\end{aligned}$$

Utilizing $a$ and $b$,

$$\begin{aligned}
&= \sum_{j\in Q} \Big( ab + (1-a)(1-b) \Big) \\
&= \sum_{j\in Q} (1 - a - b + 2ab) \\
&\leq \sum_{j\in Q} (t + 2ab)
\end{aligned}$$

For a given $0 \leq a \leq 1$, the expression $2ab$ is increasing in $b$. This means that the maximum of $2ab$ such that $0 \leq a, b \leq 1$ and $|a + b - 1| \leq t$ equals the maximum of $2ab$ such that $0 \leq a, b \leq 1$ and $a + b - 1 = t$. This is the same as the maximum of $2a(1 + t - a)$ such that $0 \leq a \leq 1$, which is $\frac{(1+t)^2}{2}$. Therefore,

$$\mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq q(\frac{(1+t)^2}{2} + t) = q(\frac{1}{2} + 2t + \frac{t^2}{2})$$

Because $t \leq 1$, the result follows. ■

We will actually need a statement similar to the last theorem, but under a slightly different probability distribution. The following claim provides what we need.

**Claim 9.3.2** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a code, and let* $Q$ *be a size* $q$ *query set an algorithm* $A$ *queries. Also assume that for every position* $j \in [m]$*,* $|\Pr_{x\in\{0,1\}^n}[x_1 = \mathbf{C}(x)_j] - \Pr_{x\in\{0,1\}^n}[x_1 \neq \mathbf{C}(x)_j]| \leq t$*. Consider two sets of vectors: the set of vectors in* $\{0,1\}^n$ *with first bit* 0 *and the set of vectors in* $\{0,1\}^n$ *with first bit* 1*. Then there exists a matching* $M$ *between those two sets such that*

$$\mathop{\mathbb{E}}_{(w_1,w_2)\in M} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq q(\frac{1}{2} + 3t)$$

**Proof:** Consider the following family of matchings, parameterized by $u \in \{0,1\}^{n-1}$

$$M_u \triangleq \Big\{(v, v+u) \mid v \in \{0,1\}^{n-1}\Big\}$$

The key property we use about this family is that it is a partition of the set $\{(w_1, w_2) \mid w_1, w_2 \in \{0,1\}^{n-1}\}$. If we consider a probability distribution in which $u$ is drawn uniformly at random from $\{0,1\}^{n-1}$ and then $(w_1, w_2)$ is drawn uniformly at random from $M_u$, then

$$\begin{aligned}\mathop{\mathbb{E}}_{u\in\{0,1\}^{n-1}} \mathop{\mathbb{E}}_{(w_1,w_2)\in M_u} & d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \\ &= \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \\ &\leq q(\frac{1}{2} + 3t) \qquad \text{by Theorem 9.3.1}\end{aligned}$$

Therefore, for at least one $u$, $\mathbb{E}_{(w_1,w_2)\in M_u}\, d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq q(\frac{1}{2}+3t)$. ■

## 9.4 q Query, Binary, Possibly Non-Linear LDCs

Now we present a correctness theorem for binary (possibly non-linear) codes.

**Theorem 9.4.1** *Let $\mathbf{C}$: $\{0,1\}^n \to \{0,1\}^m$ be a code, and let $A$ be a $q$ query recovery algorithm for it. Fix $t = o(1)$, and let $\nu \triangleq \frac{1}{.99n(1-H(\frac{1}{2}+\frac{t}{2}))}$. Then, for large enough $n$,*

$$\zeta_\delta(A) \leq 1 - .99\frac{\delta}{\sqrt{q(1+6t)}}\Big(4\delta(1-\delta)\Big)^{q(\frac{1}{4}+1.5t)} + \frac{2^{q+1}q^2}{n} + (q+1)\nu$$

**Proof:** For $i \in [n]$, define:

$$R_i \triangleq \Big\{ j \in [m] \;\Big|\; \Big| \Pr_{x\in\{0,1\}^n}[x_i = \mathbf{C}(x)_j] - \Pr_{x\in\{0,1\}^n}[x_i \neq \mathbf{C}(x)_j] \Big| > t \Big\}$$

Now consider:

$$S \triangleq \Big\{ i \in [n] \;\Big|\; |R_i| \geq \nu m \Big\}$$

Clearly $|S|\nu m \leq \sum_{i\in[n]} |R_i|$. So there exists a $j \in [m]$ belonging to at least $\nu|S|$ of the $R_i$ sets. Theorem 2 from [32] then proves that $\nu|S| \leq \frac{1}{1-H(\frac{1}{2}+\frac{t}{2})}$. Therefore, $|S| \leq \frac{1}{\nu}\frac{1}{1-H(\frac{1}{2}+\frac{t}{2})} = .99n < n$. So $\bar{S}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S}$. That is, $|R_1| < \nu m$. Consider what

happens when the recovery algorithm is tasked to find $x_1$.

Because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries exactly $q$ positions. If the algorithm ever queried fewer than $q$ positions, have it query more and ignore the additional values obtained.

Define $\gamma \triangleq \frac{|[m] \setminus R_1|}{m}$ and $\beta \triangleq \frac{\delta - \nu}{\gamma}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus R_1$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1$.

A property of $B$ that we will use later is that, for all query sets $Q$ of size $q$, value $r \in \{0,1\}$, and string $s \in \{0,1\}^q$, $\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = r] > 0$. This is because $B$ is independent of $x$. Since all of these probabilities are positive, we will be able to condition on these events properly. Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$.

For $Q$ and $r \in \{0,1\}$ having $\Pr_x[A \text{ queries } Q; x_1 = r] > 0$, define $Err_{Q,r} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq r \mid A \text{ queries } Q; x_1 = r]$. Noting that the distributions of $x$ and the internal randomness of $A$ are independent, we decompose:

$$\begin{aligned}\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] &= \sum_Q (Err_{Q,0} \Pr_x[x_1 = 0] + Err_{Q,1} \Pr_x[x_1 = 1]) \cdot \\ &\qquad \Pr[A \text{ queries } Q] \\ &= \frac{1}{2} \sum_Q (Err_{Q,0} + Err_{Q,1}) \Pr[A \text{ queries } Q]\end{aligned}$$

So let us lower bound $Err_{Q,0} + Err_{Q,1}$:

$$\begin{aligned}&Err_{Q,0} + Err_{Q,1} \\ &= \sum_{s \in \{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; x_1 = 0; (\mathbf{C}(x)+B)_Q = s] \cdot \\ &\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0] \\ &\quad + \sum_{s \in \{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; x_1 = 1; (\mathbf{C}(x)+B)_Q = s] \cdot \\ &\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1]\end{aligned}$$

The value of $x_1$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4, we can remove the conditioning on the value of

$x_1$:

$$
\begin{aligned}
&Err_{Q,0} + Err_{Q,1} \\
&\quad = \sum_{s\in\{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] \cdot \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0] \\
&\qquad + \sum_{s\in\{0,1\}^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] \cdot \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1]
\end{aligned}
$$

For ease of notation, let us make the following definition.

$$
p_s^Q \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s]
$$

This gives

$$
\begin{aligned}
&Err_{Q,0} + Err_{Q,1} \\
&\quad = \sum_{s\in\{0,1\}^q} (1 - p_s^Q) \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0] \\
&\qquad\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1]
\end{aligned}
$$

The internal randomness of $A$ is independent of $B$, and the values of the positions labeled by $Q$ are independent of whether the algorithm actually

queries $Q$. So we have

$$\begin{aligned}
Err_{Q,0} + Err_{Q,1} &= \sum_{s\in\{0,1\}^q} (1 - p_s^Q) \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \\
&\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] \\
&= \sum_{s\in\{0,1\}^q} \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] \\
&\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Big(\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]\Big) \\
&= 1 + \sum_{s\in\{0,1\}^q} p_s^Q \Big(\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]\Big)
\end{aligned}$$

This expression is smallest when $p_s^Q$ is 0 for $s$ such that $\Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] > \Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]$ and $p_s^Q$ is 1 for $s$ such that $\Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] < \Pr[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]$. Thus the last expression is lower bounded by

$$\begin{aligned}
&\geq 1 + \sum_{\substack{s\in\{0,1\}^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0]\Big)
\end{aligned}$$

where, to save space, we have used shorthand notation for specifying the last summation over $s$. Let us prove a side fact. It is clear that:

$$\begin{aligned}
\sum_{s\in\Sigma^q} \Big(\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = 1]\Big) = \\
1 - 1 = 0
\end{aligned}$$

This implies

$$\sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\Big) =$$
$$\sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=1]>\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big)$$

Therefore,

$$\sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\Big) =$$
$$\frac{1}{2}\sum_{s\in\Sigma^q} \Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big|$$

Plugging this in above,

$$Err_{Q,0} + Err_{Q,1} \geq$$
$$1 - \frac{1}{2}\sum_{s\in\{0,1\}^q} \Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big|$$

To simplify things, let us temporarily just operate on the sub-expression:

$$\begin{aligned}
\Delta &\triangleq \sum_{s\in\{0,1\}^q} \Big| \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] \Big| \\
&= \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 1w] \Pr_x[x_2x_3...x_n = w \mid x_1 = 1] \\
&\qquad - \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w] \Pr_x[x_2x_3...x_n = w \mid x_1 = 0] \Big| \\
&= \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 1w] \Pr_x[x_2x_3...x_n = w] - \\
&\qquad \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w] \Pr_x[x_2x_3...x_n = w] \Big| \\
&= \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Pr_x[x_2x_3...x_n = w] \Big( \Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \\
&\qquad\qquad \Pr_B[B_Q = s - \mathbf{C}(0w)_Q] \Big) \Big| \\
&= \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \sum_{w\in\{0,1\}^{n-1}} \Big( \Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q] \Big) \Big|
\end{aligned}$$

At this point we apply Claim 9.3.2 to find a special matching $M$ between the set of vectors in $\{0,1\}^n$ with first bit 0 and the set of vectors in $\{0,1\}^n$

with first bit 1. Using $M$,

$$\Delta = \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \Big| \sum_{(w_1,w_2)\in M} \Big( \Pr_B[B_Q = s - \mathbf{C}(1w_1)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w_2)_Q] \Big) \Big|$$

$$\leq \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \sum_{(w_1,w_2)\in M} \Big| \Pr_B[B_Q = s - \mathbf{C}(1w_1)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w_2)_Q] \Big|$$

$$= \frac{1}{2^{n-1}} \sum_{(w_1,w_2)\in M} \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s - \mathbf{C}(1w_1)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w_2)_Q] \Big|$$

switch summations

$$= \frac{1}{2^{n-1}} \sum_{(w_1,w_2)\in M} \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q] \Big|$$

$s - \mathbf{C}(1w_1)_Q$ relabeled $s$

For each $(w_1, w_2) \in M$, we will bound

$$\Delta_{w_1,w_2} \triangleq \sum_{s\in\{0,1\}^q} \Big| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q] \Big|$$

For a given $s$, $\Pr_B[B_Q = s]$ is a function of only the number of 1's in $s$. Let $\mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q$ have $a(w_1, w_2)$ 1's. To simplify notation, we will temporarily refer to $a(w_1, w_2)$ as $a$. Also, let $b$ be the number of positions where $\mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q$ and $s$ both equal 1; and let $c$ be the number of positions where $\mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q$ equals 0 but $s$ equals 1. Therefore, by Lemmas 5.2.2 and 5.2.3 (for large enough $m$), we have

$$\Delta_{w_1,w_2} \leq$$
$$\sum_{b=0}^{a} \binom{a}{b} \sum_{c=0}^{q-a} \binom{q-a}{c} \Big( \Big| \beta^{b+c}(1-\beta)^{q-b-c} - \beta^{a-b+c}(1-\beta)^{q-a+b-c} \Big| + \frac{3q^2}{m} \Big)$$

Simplifying, we have

$$\begin{aligned}
&\Delta_{w_1,w_2} \\
&= \sum_{b=0}^{a} \binom{a}{b} \sum_{c=0}^{q-a} \binom{q-a}{c} \left| \beta^{b+c}(1-\beta)^{q-b-c} - \beta^{a-b+c}(1-\beta)^{q-a+b-c} \right| + \frac{3 * 2^q q^2}{m} \\
&= \sum_{b=0}^{a} \binom{a}{b} \Big( \sum_{c=0}^{q-a} \binom{q-a}{c} \beta^c (1-\beta)^{q-a-c} \Big) \left| \beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b \right| + \\
&\qquad\qquad \frac{3 * 2^q q^2}{m} \\
&= \sum_{b=0}^{a} \binom{a}{b} \left| \beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b \right| + \frac{3 * 2^q q^2}{m}
\end{aligned}$$

$\delta < \frac{1}{2}$ must hold for $A$ to have nontrivial correctness. Hence $\delta < 1 - \delta$ and $\beta < 1 - \beta$. Therefore, the expression inside of the absolute value is positive if and only if $b < \frac{a}{2}$. Also note that the value of the expression inside of the absolute value, for a given $b$, has the opposite sign of that same expression when $b$ is replaced by $a - b$. Using these facts, we have

$$\Delta_{w_1,w_2} = 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \Big( \beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b \Big) + \frac{3 * 2^q q^2}{m}$$

Because $\sum_{b=0}^{a} \binom{a}{b} \beta^b (1-\beta)^{a-b} = (1-\beta+\beta)^a = 1$, the last line equals

$$
\begin{aligned}
&= 2\Big(1 - \sum_{b=\lfloor (a+1)/2 \rfloor}^{a} \binom{a}{b} \beta^b (1-\beta)^{a-b} - \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \beta^{a-b} (1-\beta)^b \Big) + \frac{3 * 2^q q^2}{m} \\
&\leq 2\Big(1 - 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \beta^{a-b} (1-\beta)^b \Big) + \frac{3 * 2^q q^2}{m} \\
&\qquad\qquad \text{interchanging } b \text{ and } a-b \text{ in second term} \\
&< 2 - 4 \binom{a}{\lfloor (a-1)/2 \rfloor} \beta^{\lceil (a+1)/2 \rceil} (1-\beta)^{\lfloor (a-1)/2 \rfloor} + \frac{3 * 2^q q^2}{m} \\
&< 2 - 3.96 \binom{a}{\lfloor a/2 \rfloor} \beta^{a/2+1} (1-\beta)^{a/2} + \frac{3 * 2^q q^2}{m}
\end{aligned}
$$

Note that for large $\delta$ and thus $\beta$, stronger upper bounds than the ones used in the last equation block could have been used instead. The last line holds because, for $n$ large enough, the ratio between the central binomial coefficient and the binomial coefficient whose lower register is one less is less than 1 plus any constant. Stirling's formula shows that, for any $z > 1$, $\binom{2z}{z} \geq \frac{4^z}{2\sqrt{z}}$. Combining these facts gives

$$
\begin{aligned}
\Delta_{w_1,w_2} &< 2 - 3.96 \frac{2^a}{\sqrt{2a}} \beta^{a/2+1} (1-\beta)^{a/2} + \frac{3 * 2^q q^2}{m} \\
&= 2 - 3.96 \frac{\beta}{\sqrt{2a}} (4\beta(1-\beta))^{a/2} + \frac{3 * 2^q q^2}{m}
\end{aligned}
$$

It is clear that $\Delta = \frac{1}{2^{n-1}} \sum_{(w_1,w_2) \in M} \Delta_{w_1,w_2}$. Fact 9.4.2 shows that the function $\phi(a) = \frac{1}{\sqrt{2a}} (4\beta(1-\beta))^{a/2}$ is convex. Claim 9.3.2 shows that $\frac{1}{2^{n-1}} \sum_{(w_1,w_2) \in M} a(w_1, w_2) \leq q(\frac{1}{2} + 3t)$. Jensen's inequality states that, for a convex function $f$, the average of $f$ applied to two different inputs is greater

than $f$ applied to the average of those inputs. Thus,

$$\Delta < 2 - 3.96\frac{\beta}{\sqrt{q(1+6t)}}(4\beta(1-\beta))^{q(\frac{1}{4}+1.5t)} + \frac{3*2^q q^2}{m}$$

Therefore, we have

$$\begin{aligned} Err_{Q,0} + Err_{Q,1} &\geq 1 - \frac{1}{2}\Big(2 - 3.96\frac{\beta}{\sqrt{q(1+6t)}}(4\beta(1-\beta))^{q(\frac{1}{4}+1.5t)} + \frac{3*2^q q^2}{m}\Big) \\ &= 1.98\frac{\beta}{\sqrt{q(1+6t)}}\Big(4\beta(1-\beta)\Big)^{q(\frac{1}{4}+1.5t)} - \frac{3*2^q q^2}{m} \end{aligned}$$

Remember that $\beta = \frac{\delta-\nu}{\gamma}$, first note that $1.98\frac{\beta}{\sqrt{q(1+6t)}}\Big(4\beta(1-\beta)\Big)^{q(\frac{1}{4}+1.5t)}$ is strictly increasing in $\beta$. Therefore, we can lower bound $1.98\frac{\beta}{\sqrt{q(1+6t)}}\Big(4\beta(1-\beta)\Big)^{q(\frac{1}{4}+1.5t)}$ evaluated at $\beta = \frac{\delta-\nu}{\gamma}$ with $1.98\frac{\hat{\beta}}{\sqrt{q(1+6t)}}\Big(4\hat{\beta}(1-\hat{\beta})\Big)^{q(\frac{1}{4}+1.5t)}$ evaluated at $\hat{\beta} = \delta - \nu$. Thus, we have

$$\begin{aligned} Err_{Q,0} + Err_{Q,1} &\geq 1.98\frac{\delta-\nu}{\sqrt{q(1+6t)}}\Big(4(\delta-\nu)(1-(\delta-\nu))\Big)^{q(\frac{1}{4}+1.5t)} - \frac{3*2^q q^2}{m} \\ &\geq 1.98\frac{\delta-\nu}{\sqrt{q(1+6t)}}\Big(4(\delta-\nu)(1-\delta)\Big)^{q(\frac{1}{4}+1.5t)} - \frac{3*2^q q^2}{m} \\ &\geq 1.98\frac{\delta-\nu}{\sqrt{q(1+6t)}}\Big(4\delta(1-\delta) - 4\nu\Big)^{q(\frac{1}{4}+1.5t)} - \frac{3*2^q q^2}{m} \\ &\geq 1.98\frac{\delta}{\sqrt{q(1+6t)}}\Big(4\delta(1-\delta)\Big)^{q(\frac{1}{4}+1.5t)} - (2q+2)\nu - \frac{3*2^q q^2}{m} \end{aligned}$$

The final result is obtained by utilizing the lower bound of Katz and Trevisan [32]. This implies, for large enough $n$, $m > n$. ■

**Fact 9.4.2** *Given that $\beta < \frac{1}{2}$, the function $\phi(a) = \frac{1}{\sqrt{2a}}(4\beta(1-\beta))^{a/2}$ is convex for positive $a$.*

**Proof:** Note that $\phi(a)$ can be rewritten as $\chi(b) \triangleq \frac{\gamma^b}{2\sqrt{b}}$ where $b \triangleq a/2$ and $\gamma \triangleq 4\beta(1-\beta) < 1$.

The first derivative of $\chi(b)$, with respect to $b$, is

$$\gamma^b(-\frac{1}{4}b^{-3/2} + \frac{1}{2}b^{-1/2}\log\gamma)$$

The second derivative of $\chi(b)$ is

$$\gamma^b(\frac{3}{8}b^{-5/2} - \frac{1}{2}b^{-3/2}\log\gamma + \frac{1}{2}b^{-1/2}(\log\gamma)^2)$$

Because $\log\gamma < 0$, $\gamma > 0$, and $b > 0$, each term of the last line is positive. So the last line as a whole is positive. Also note that these first and second derivatives are continuous over the domain of positive $b$. So $\chi(b)$ is convex. Because the derivative of $b$ with respect to $a$ is $\frac{1}{2} > 0$, then, by the chain rule, $\phi(a)$ is convex as well. ■

## 9.5 q Query, Possibly Non-Linear LDCs Over Any Field

**Theorem 9.5.1** *Let* $\mathbf{C}$*:* $\Sigma^n \to \Sigma^m$ *be a code, and let* $A$ *be a* $q$ *query recovery algorithm for it. Then, for large enough* $m$*,*

$$\zeta^\star_\delta(A) \leq 1 - 2\frac{|\Sigma|-1}{|\Sigma|}\binom{q}{\lfloor (q-1)/2 \rfloor}(\frac{\delta}{|\Sigma|-1})^{\lceil (q+1)/2 \rceil}(1-\delta)^{\lfloor (q-1)/2 \rfloor} + \frac{2^q q^2}{m}$$

**Proof:** Without loss of generality, let $\Sigma = \{0, 1, ...|\Sigma|-1\}$. We will use modular arithmetic (modulo $|\Sigma|$) on $\Sigma$. Also, define $\hat{\Sigma}$ as the nonzero elements

of $\Sigma$.

Without loss of generality, let $x_1$ be the input bit $A$ has been tasked to find.

Let $x$ be drawn uniformly at random from $\Sigma^n$. The adversary will choose $B \subset [m]$ uniformly at random from all subsets of $[m]$ having size $\delta m$. The adversary will corrupt each position in $B$ by adding to it a value from $\hat{\Sigma}$ uniformly at random. A property of $B$ that we use later is, for all query sets $Q$ of size $q$, value $r \in \Sigma$, and string $s \in \Sigma^q$, $\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = r] > 0$. This is because $B$ is independent of $x$. Since all of these probabilities are positive, we will be able to condition on these events properly.

For $Q$ and $r \in \Sigma$ having $\Pr_x[A \text{ queries } Q; x_1 = r] > 0$, define $Err_{Q,r} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq r \mid A \text{ queries } Q; x_1 = r]$. Noting that the distributions of $x$ and the internal randomness of $A$ are independent, we decompose:

$$\begin{aligned}\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] &= \sum_Q \sum_{r \in \Sigma} Err_{Q,r} \Pr_x[x_1 = r] \Pr[A \text{ queries } Q] \\ &= \frac{1}{|\Sigma|} \sum_Q \sum_{r \in \Sigma} Err_{Q,t} \Pr[A \text{ queries } Q]\end{aligned}$$

So let us lower bound $\sum_{r\in\Sigma} Err_{Q,r}$:

$$\sum_{r\in\Sigma} Err_{Q,r} =$$
$$\sum_{r\in\Sigma}\sum_{s\in\Sigma^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1)\neq r \mid A \text{ queries } Q; x_1 = r; (\mathbf{C}(x)+B)_Q = s]\cdot$$
$$\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = r]$$

The value of $x_1$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4, we can remove the conditioning on the value of $x_1$:

$$\sum_{r\in\Sigma} Err_{Q,r} = \sum_{r\in\Sigma}\sum_{s\in\Sigma^q} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1)\neq r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s]\cdot$$
$$\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = r]$$

For ease of notation, let us make the following definition.

$$p_s^Q(r) \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = r \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s]$$

such that, for every $Q$ and $s$, $\sum_{r\in\Sigma} p_s^Q(r) = 1$. This gives

$$\sum_{r\in\Sigma} Err_{Q,r}$$
$$= \sum_{r\in\Sigma}\sum_{s\in\Sigma^q}\Big(1-p_s^Q(r)\Big)\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = r]$$
$$= \sum_{s\in\Sigma^q}\Big(\sum_{r\in\hat{\Sigma}} p_s^Q(r)\Big)\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0]+$$
$$\sum_{r\in\hat{\Sigma}}\sum_{s\in\Sigma^q}\Big(1-p_s^Q(r)\Big)\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = r]$$

The internal randomness of $A$ is independent of $B$, and the values of the positions labeled by $Q$ are independent of whether the algorithm actually queries $Q$. So we have

$$\begin{aligned}
\sum_{r\in\Sigma} Err_{Q,r} &= \sum_{r\in\hat{\Sigma}}\Big(\sum_{s\in\Sigma^q} p_s^Q \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] \\
&\qquad + \sum_{s\in\Sigma^q}(1-p_s^Q(r))\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r]\Big) \\
&= \sum_{r\in\hat{\Sigma}}\Big(\sum_{s\in\Sigma^q}\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r] \\
&\qquad + \sum_{s\in\Sigma^q} p_s^Q(r)\Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r]\Big)\Big) \\
&= |\Sigma| - 1 + \sum_{r\in\hat{\Sigma}}\sum_{s\in\Sigma^q} p_s^Q(r)\Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \\
&\qquad\qquad \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r]\Big)
\end{aligned}$$

This expression is smallest when $p_s^Q(r)$ is 0 for $s$ such that $\Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] > \Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r]$ and $p_s^Q(r)$ is 1 for $s$ such that $\Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] < \Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r]$. Thus the last expression is lower bounded by

$$\begin{aligned}
&\geq |\Sigma| - 1 + \sum_{r\in\hat{\Sigma}} \sum_{\substack{s\in\Sigma^q \\ \Pr[s|x_1=0]<\Pr[s|x_1=r]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] \\
&\qquad\qquad - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r]\Big)
\end{aligned}$$

where, to save space, we have used shorthand notation for specifying the last summation over $s$. Let us prove a side fact. It is clear that:

$$\sum_{s\in\Sigma^q}\Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=0]-\Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=r]\Big)= 1-1=0$$

This implies

$$\sum_{\substack{s\in\Sigma^q\\ \Pr[s|x_1=r]<\Pr[s|x_1=0]}}\Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=0]- \Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=r]\Big)=$$
$$\sum_{\substack{s\in\Sigma^q\\ \Pr[s|x_1=r]>\Pr[s|x_1=0]}}\Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=r]- \Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=0]\Big)$$

Therefore,

$$\sum_{\substack{s\in\Sigma^q\\ \Pr[s|x_1=r]<\Pr[s|x_1=0]}}\Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=0]- \Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=r]\Big)=$$
$$\frac{1}{2}\sum_{s\in\Sigma^q}\Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=r]- \Pr_{x,B}[(\mathbf{C}(x)+B)_Q=s \mid x_1=0]\Big|$$

Plugging this in above,

$$\sum_{r\in\Sigma} Err_{Q,r} \geq |\Sigma| - 1 - \frac{1}{2}\sum_{r\in\hat{\Sigma}}\sum_{s\in\Sigma^q}\Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big|$$

To simplify things, let us temporarily just operate on the sub-expression:

$$\begin{aligned}
\Delta_r &\triangleq \sum_{s\in\Sigma^q}\Big|\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = r] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big| \\
&= \sum_{s\in\Sigma^q}\Big|\sum_{w\in\Sigma^{n-1}}\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = rw]\Pr_x[x_2x_3...x_n = w \mid x_1 = r] - \\
&\qquad \sum_{w\in\Sigma^{n-1}}\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w]\Pr_x[x_2x_3...x_n = w \mid x_1 = 0]\Big| \\
&= \sum_{s\in\Sigma^q}\Big|\sum_{w\in\Sigma^{n-1}}\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = rw]\Pr_x[x_2x_3...x_n = w] - \\
&\qquad \sum_{w\in\Sigma^{n-1}}\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w]\Pr_x[x_2x_3...x_n = w]\Big| \\
&= \sum_{s\in\Sigma^q}\Big|\sum_{w\in\Sigma^{n-1}}\Pr_x[x_2x_3...x_n = w]\Big(\Pr_B[B_Q = s - \mathbf{C}(rw)_Q] - \\
&\qquad\qquad \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\Big)\Big| \\
&= \frac{1}{|\Sigma|^{n-1}}\sum_{s\in\Sigma^q}\Big|\sum_{w\in\Sigma^{n-1}}\Big(\Pr_B[B_Q = s - \mathbf{C}(rw)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\Big)\Big| \\
&\leq \frac{1}{|\Sigma|^{n-1}}\sum_{s\in\Sigma^q}\sum_{w\in\Sigma^{n-1}}\Big|\Pr_B[B_Q = s - \mathbf{C}(rw)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\Big| \\
&= \frac{1}{|\Sigma|^{n-1}}\sum_{w\in\Sigma^{n-1}}\sum_{s\in\Sigma^q}\Big|\Pr_B[B_Q = s - \mathbf{C}(rw)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\Big| \\
&\qquad\qquad \text{switch summations} \\
&= \frac{1}{|\Sigma|^{n-1}}\sum_{w\in\Sigma^{n-1}}\sum_{s\in\Sigma^q}\Big|\Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(rw)_Q - \mathbf{C}(0w)_Q]\Big| \\
&\qquad\qquad s - \mathbf{C}(rw)_Q \text{ relabeled } s
\end{aligned}$$

For each $r \in \hat{\Sigma}$ and $w \in \Sigma^{n-1}$, we will bound

$$\Delta_{r,w} \triangleq \sum_{s \in \Sigma^q} \Big| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(rw)_Q - \mathbf{C}(0w)_Q] \Big|$$

Now will be the first time we use a fact about the distribution of $B$, excluding when we used that $B$ is independent of $A$. For a given $s$, $\Pr_B[B_Q = s]$ is a function of only the number of nonzero elements in $s$. Let $\mathbf{C}(rw)_Q - \mathbf{C}(0w)_Q$ have $a$ nonzero elements. Also, let $b$ be the number of positions where $\mathbf{C}(rw)_Q - \mathbf{C}(0w)_Q$ and $s$ are both nonzero; and let $c$ be the number of positions where $\mathbf{C}(rw)_Q - \mathbf{C}(0w)_Q$ equals 0 but $s$ is nonzero. Therefore, by Lemmas 5.2.2 and 5.2.3 (for large enough $m$), we have

$$\Delta_{r,w} \leq$$
$$\sum_{b=0}^{a} \binom{a}{b} \sum_{c=0}^{q-a} \binom{q-a}{c} \Big( \Big| \frac{\delta^{b+c}(1-\delta)^{q-b-c}}{(|\Sigma|-1)^{b+c}} - \frac{\delta^{a-b+c}(1-\delta)^{q-a+b-c}}{(|\Sigma|-1)^{a-b+c}} \Big| + \frac{3q^2}{m} \Big)$$

Simplifying, we have

$$\leq \sum_{b=0}^{a} \binom{a}{b} \sum_{c=0}^{q-a} \binom{q-a}{c} \frac{1}{(|\Sigma|-1)^c} \Big| \frac{\delta^{b+c}(1-\delta)^{q-b-c}}{(|\Sigma|-1)^b} - \frac{\delta^{a-b+c}(1-\delta)^{q-a+b-c}}{(|\Sigma|-1)^{a-b}} \Big| + \frac{3 * 2^q q^2}{m}$$
$$= \sum_{b=0}^{a} \binom{a}{b} \Big( \sum_{c=0}^{q-a} \binom{q-a}{c} \big(\frac{\delta}{|\Sigma|-1}\big)^c (1-\delta)^{q-a-c} \Big) \Big| \big(\frac{\delta}{|\Sigma|-1}\big)^b (1-\delta)^{a-b} - \big(\frac{\delta}{|\Sigma|-1}\big)^{a-b} (1-\delta)^b \Big| + \frac{3 * 2^q q^2}{m}$$
$$\leq \sum_{b=0}^{a} \binom{a}{b} \Big| \big(\frac{\delta}{|\Sigma|-1}\big)^b (1-\delta)^{a-b} - \big(\frac{\delta}{|\Sigma|-1}\big)^{a-b} (1-\delta)^b \Big| + \frac{3 * 2^q q^2}{m}$$

$\delta < \frac{1}{|\Sigma|}$ must hold for $A$ to have nontrivial correctness. Hence $\frac{\delta}{|\Sigma|-1} < 1-\delta$. Therefore, the expression inside of the absolute value is positive if and only if $b < \frac{a}{2}$. Also note that the value of the expression inside of the absolute value, for a given $b$, has the opposite sign of that same expression when $b$ is replaced by $a-b$. Using these facts, we have

$$= 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \Big( (\frac{\delta}{|\Sigma|-1})^b (1-\delta)^{a-b} - (\frac{\delta}{|\Sigma|-1})^{a-b} (1-\delta)^b \Big) + \frac{3 * 2^q q^2}{m}$$

Because $\sum_{b=0}^{a} \binom{a}{b} (\frac{\delta}{|\Sigma|-1})^b (1-\delta)^{a-b} = (1-\delta+\frac{\delta}{|\Sigma|-1})^a \leq 1$, the last line is less than or equal

$$\leq 2 \Big( 1 - \sum_{b=\lfloor (a+1)/2 \rfloor}^{a} \binom{a}{b} (\frac{\delta}{|\Sigma|-1})^b (1-\delta)^{a-b} - \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} (\frac{\delta}{|\Sigma|-1})^{a-b} (1-\delta)^b \Big) + \frac{3 * 2^q q^2}{m}$$

$$\leq 2 \Big( 1 - 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} (\frac{\delta}{|\Sigma|-1})^{a-b} (1-\delta)^b \Big) + \frac{3 * 2^q q^2}{m}$$

interchanging $b$ and $a-b$ in second term

$$< 2 - 4 \binom{a}{\lfloor (a-1)/2 \rfloor} (\frac{\delta}{|\Sigma|-1})^{\lceil (a+1)/2 \rceil} (1-\delta)^{\lfloor (a-1)/2 \rfloor} + \frac{3 * 2^q q^2}{m}$$

$$\leq 2 - 4 \binom{q}{\lfloor (q-1)/2 \rfloor} (\frac{\delta}{|\Sigma|-1})^{\lceil (q+1)/2 \rceil} (1-\delta)^{\lfloor (q-1)/2 \rfloor} + \frac{3 * 2^q q^2}{m}$$

Note that for large $\delta$, stronger upper bounds than the ones used in the last equation block could have been used instead.

So, all the $\Delta_{r,w}$'s have this same upper bound. Therefore, we have

$$\sum_{r\in\Sigma} Err_{Q,r} \geq |\Sigma| - 1 - \frac{|\Sigma|-1}{2}\Bigg(2 - 4\binom{q}{\lfloor (q-1)/2 \rfloor}\delta^{\lceil (q+1)/2\rceil}(1-\delta)^{\lfloor (q-1)/2\rfloor} + \frac{3*2^q q^2}{m}\Bigg)$$

$$= 2(|\Sigma|-1)\binom{q}{\lfloor (q-1)/2 \rfloor}\left(\frac{\delta}{|\Sigma|-1}\right)^{\lceil (q+1)/2\rceil}(1-\delta)^{\lfloor (q-1)/2\rfloor} - \frac{3(|\Sigma|-1)2^{q-1}q^2}{m}$$

■

We note that, for binary codes, this correctness bound is tight up to a constant factor. This can be seen by the following. Take the code with all positions equalling $x_1$. While algorithms querying this code cannot return anything other than the first input position with probability greater than random guessing, there is an algorithm that is very good at returning that first bit. Specifically, for any $q$, there is a $q$ query algorithm for this code that achieves correctness

$$\zeta_\delta \geq 1 - 2^q(1-\delta)^{\lfloor \frac{q}{2}\rfloor}\delta^{\lceil \frac{q}{2}\rceil}q \qquad \text{assuming } \delta \leq \frac{1}{2}$$

when tasked to recover input position 1. This algorithm queries $q$ random positions and takes the majority vote of the answers it receives. See the proof of Theorem 12.2 for a derivation of the bound.

# Chapter 10

# Precise Bounds on Correctness for Two and Three Query LDCs

The correctness bounds in the last chapter are interesting for large $q$, but they are not tight for small $q$. To address this, this section provides several correctness bounds on the performance of algorithms allowing just two or three queries.

## 10.1 Two Query, Binary, Linear LDCs

**Claim 10.1** *Let* $\mathbf{C}$*:* $\{0,1\}^n \rightarrow \{0,1\}^m$ *be a linear code. For any two query LDC recovery algorithm A,* $\zeta_\delta(A) \leq \max(\frac{1}{2}, 1 - 2\delta + \frac{2}{n})$.

**Proof:** By Claim 2.2, there exists a code **C'** with no codeword position identically zero and an algorithm $A'$ operating on it such that, for any $\delta$, $\zeta_\delta(A) \leq \zeta_\delta(A')$. In this proof, we will henceforth work exclusively with **C'** and $A'$, but, for simplicity, drop the primes from notation.

For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \mathbf{C}(x)_j = x_i \right\}$$

So there exists at least one $i$ such that $|R_i| \leq \frac{m}{n}$. Without loss of generality, assume $|R_1| \leq \frac{m}{n}$. Consider what happens when the recovery algorithm is tasked to find $x_1$. Recall the notation we defined earlier, applicable because $\mathbf{C}$ is linear: for $j \in [m]$, $a_j \in \{0,1\}^n$ is the vector satisfying $\forall x \in \{0,1\}^n$, $\mathbf{C}_j(x) = a_j \cdot x$. Using this, define

$$S \triangleq \left\{ j \in [m] \mid (a_j)_1 \neq 0 \right\}$$

Define $T$ as whichever of $S$ and $\bar{S}$ (the complement of $S$) has smaller size. If they have the same size, $T$ can be either set. Clearly $|T| \leq \frac{m}{2}$.

Define $\gamma \triangleq \frac{|T|}{m}$ and $\beta \triangleq \min(\frac{\delta - \frac{|R_1|}{m}}{\gamma}, \frac{1}{2})$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset T$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. We also note that, because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries two positions. If the algorithm ever queried fewer than two positions, have it query more and ignore the additional values obtained. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1$.

Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1]$$
$$= \sum_{Q \subset [m], |Q|=2} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$. Define $Err_Q \triangleq$
$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound $Err_Q$ depending on all the different possibilities for $Q$ for which $\Pr[A \text{ queries } Q] > 0$. First we give some notation.

Write $Q = \{j_1, j_2\}$. For $Q$ and $a, b \in \{0,1\}$ such that
$\Pr_{x,B}[A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = ab] > 0$, define

$$p_{ab}^Q \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = ab]$$

For simplicity, let us define the following notation. For a given $S \subseteq Q$,

$$q_{ab}^{Q,k}(S) \triangleq \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = ab \mid A \text{ queries } Q; |B \cap S| = k]$$

On an intuitive level, the cases below have similarities to each other, but they use different $S$ in their analyses. Here are the possibilities for $Q$:

- $e_1 \notin \text{span}\{a_{j_1}, a_{j_2}\}$: By Theorem 5.1.1, $Err_Q \geq \frac{1}{2}$. Because $\beta \leq \frac{1}{2}$, $Err_Q \geq \beta$ as well.

- $e_1 \in \text{span}\{a_{j_1}, a_{j_2}\}$: Exactly one bit of $Q$ must be in $T$ – assume it is $j_1$. We can decompose $Err_Q$ into

$$\begin{aligned} Err_Q &= \sum_{k=0}^{1} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k] \cdot \\ &\qquad \Pr_{B}[|B \cap \{j_1\}| = k \mid A \text{ queries } Q] \\ &= \sum_{k=0}^{1} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k] \cdot \\ &\qquad \Pr_{B}[|B \cap \{j_1\}| = k] \end{aligned}$$

Note that for any $Q$, $j_1 \in Q$, and $0 \leq k \leq 1$, the events $A$ queries $Q$ and $|B \cap \{j_1\}| = k$ are independent. So for any $Q$, $j_1 \in Q$, and $0 \leq k \leq 1$, $\Pr[A \text{ queries } Q; |B \cap \{j_1\}| = k] > 0$. Thus, above we are conditioning on events with nonzero probability. The second equality above also holds because of the independence of $A$ queries $Q$ and $|B \cap \{j_1\}| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k]$. We can further decompose $Err_{Q,k}$ into

$$\begin{aligned} Err_{Q,k} = \sum_{a,b} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k; \\ (\mathbf{C}(x) + B)_Q = ab] \cdot \\ \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = ab \mid A \text{ queries } Q; |B \cap \{j_1\}| = k] \end{aligned}$$

Since neither bit is in $R_1$ but $e_1$ is in the span of both bits, the sum of the two bits (when uncorrupted) is $x_1$. (In this proof, additions involving codeword bits are implicitly modulo 2.) So $a + b = x_1 + |B \cap \{j_1\}|$, and

the above becomes:

$$
\begin{aligned}
Err_{Q,k} = & \sum_{\substack{a,b \\ a+b=k}} q_{ab}^{Q,k}(\{j_1\})\cdot \\
& \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k; (\mathbf{C}(x)+B)_Q = ab] \\
& \quad + \sum_{\substack{a,b \\ a+b=1+k}} q_{ab}^{Q,k}(\{j_1\})\cdot \\
& \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k; (\mathbf{C}(x)+B)_Q = ab]
\end{aligned}
$$

The event $|B \cap \{j_1\}| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$
\begin{aligned}
& Err_{Q,k} = \\
& \sum_{\substack{a,b \\ a+b=k}} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = ab] q_{ab}^{Q,k}(\{j_1\}) + \\
& \sum_{\substack{a,b \\ a+b=1+k}} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = ab] q_{ab}^{Q,k}(\{j_1\})
\end{aligned}
$$

This means,

$$
Err_{Q,k} = \sum_{\substack{a,b \\ a+b=k}} (1 - p_{ab}^{Q}) q_{ab}^{Q,k}(\{j_1\}) + \sum_{\substack{a,b \\ a+b=1+k}} p_{ab}^{Q} q_{ab}^{Q,k}(\{j_1\})
$$

The two query bits cannot be equal because one is in $T$ and one is not. Since neither is 0, they are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x)+B)_{j_1}$ and $(\mathbf{C}(x)+B)_{j_2}$ are two independent, uniformly random bits. Thus, $\forall k, a, b$: $q_{ab}^{Q,k}(\{j_1\})$

$= \frac{1}{4}$. So, in the $k = 0$ case,

$$Err_{Q,0} = \Big(p^Q_{01} + p^Q_{10} + (1 - p^Q_{00}) + (1 - p^Q_{11})\Big)/4$$

For simplicity, define $P_Q \triangleq \Big(p^Q_{01} + p^Q_{10} + (1 - p^Q_{00}) + (1 - p^Q_{11})\Big)/4$. On the other hand, in the $k = 1$ case,

$$Err_{Q,1} = \Big((1 - p^Q_{01}) + (1 - p^Q_{10}) + p^Q_{00} + p^Q_{11}\Big)/4 = 1 - P_Q$$

The probability that $j_1$ was corrupted is $\beta$. Combining everything, we find

$$Err_Q = (1 - \beta)P_Q + \beta(1 - P_Q) = \beta + (1 - 2\beta)P_Q$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq \beta$.

Since for all $Q$, $Err_Q \geq \beta$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq \beta$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq \beta$ (where the probability is only over the internal coin flips of $A$). When $\beta = \frac{1}{2}$, we are done. Otherwise $\beta = \frac{\delta - \frac{|R_1|}{m}}{\gamma} = \frac{\delta - \frac{1}{n}}{\gamma}$, and we know $\gamma \leq \frac{1}{2}$. In this case $\beta \geq 2\delta - \frac{2}{n}$. Combining these two possibilities gives the result. ■

## 10.2 Three Query, Binary, Linear LDCs

We can extend this proof for $q = 3$:

**Claim 10.2** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a linear code with* $n$ *large enough. For any three query LDC recovery algorithm* $A$*:*

$$\zeta_\delta(A) \leq \begin{cases} 1 - 2\delta(1-\delta) + \frac{46}{n} & \delta \leq \frac{1}{2} \\ \frac{1}{2} & \delta \geq \frac{1}{2} \end{cases}$$

**Proof:** By Claim 2.2, there exists a code **C'** with no codeword position identically zero and an algorithm $A'$ operating on it such that, for any $\delta$, $\zeta_\delta(A) \leq \zeta_\delta(A')$. In this proof, we will henceforth work exclusively with **C'** and $A'$, but, for simplicity, drop the primes from notation.

For $i \in [n]$, define

$$R_i \triangleq \left\{ j \in [m] \mid \mathbf{C}(x)_j = x_i \right\}$$

So there exists at least one $i$ such that $|R_i| \leq \frac{m}{n}$. Without loss of generality, assume $|R_1| \leq \frac{m}{n}$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Define $\gamma \triangleq \frac{|[m] \setminus R_1|}{m}$ and $\beta \triangleq \frac{\delta - \frac{|R_1|}{m}}{\gamma} = \frac{\delta - \frac{|R_1|}{m}}{1 - \frac{|R_1|}{m}}$. Note that if $\delta \geq \frac{1}{2}$, Claim 4.1 already gives the result. So we can assume $\delta \leq \frac{1}{2}$, and therefore $\beta \leq \frac{1}{2}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus R_1$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of $A$. (For emphasis, the adversary chooses $B_1$ to always have the same size; but, for $B_2$, it chooses

whether to include each member of $R_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x) + B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. We also note that, because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries three positions. If the algorithm ever queried fewer than three positions, have it query more and ignore the additional values obtained. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1$.

Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1]$$
$$= \sum_{Q \subset [m], |Q|=3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Note that probability expressions involving $A$ are also implicitly over the internal randomness of $A$. Define $Err_Q \triangleq$
$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound $Err_Q$ depending on all the different possibilities for $Q$ for which $\Pr[A \text{ queries } Q] > 0$. First we give some notation.

Write $Q = \{j_1, j_2, j_3\}$. For $Q$ and $a, b, c \in \{0, 1\}$ such that
$\Pr_{x,B}[A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abc] > 0$, define

$$p^Q_{abc} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abc]$$

For simplicity, let us define the following notation. For a given $S \subseteq Q$,

$$q_{abc}^{Q,k}(S) \triangleq \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = abc \mid A \text{ queries } Q; |B \cap S| = k]$$

On an intuitive level, the cases below have similarities to each other, but they use different $S$ in their analyses. Here are the possibilities for $Q$:

- $e_1 \notin \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$: By Theorem 5.1.1, $Err_Q \geq \frac{1}{2}$. Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 2\beta(1-\beta)$ as well.

- $e_1 \in \text{span}\{a_{j_1}, a_{j_2}, a_{j_3}\}$; but $e_1$ is not in the span of any two of those vectors taken by themselves: We can decompose $Err_Q$ into

$$\begin{aligned} Err_Q = \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k] \cdot \\ \Pr_B[|B \cap Q| = k \mid A \text{ queries } Q] \\ = \sum_{k=0}^{3} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k] \cdot \\ \Pr_B[|B \cap Q| = k] \end{aligned}$$

  Note that for any $Q$ and $0 \leq k \leq 3$, the events $A$ queries $Q$ and $|B \cap Q| = k$ are independent. So for any $Q$ and $0 \leq k \leq 3$, $\Pr[A \text{ queries } Q; |B \cap Q| = k] > 0$. Thus, above we are conditioning on events with nonzero probability. The second equality above also holds because of the independence of $A$ queries $Q$ and $|B \cap Q| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k]$. We can

further decompose $Err_{Q,k}$ into

$$Err_{Q,k} = \sum_{a,b,c} q_{abc}^{Q,k}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abc]$$

The sum of the three bits (when uncorrupted) is $x_1$. (In this proof, additions involving codeword bits are implicitly modulo 2.) So $a+b+c$ $= x_1 + |B \cap Q|$, and the above becomes:

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b+c=k}} q_{abc}^{Q,k}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abc]$$
$$+ \sum_{\substack{a,b,c \\ a+b+c=1+k}} q_{abc}^{Q,k}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap Q| = k; (\mathbf{C}(x)+B)_Q = abc]$$

The event $|B \cap Q| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b+c=k}} q_{abc}^{Q,k}(Q) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]$$
$$+ \sum_{\substack{a,b,c \\ a+b+c=1+k}} q_{abc}^{Q,k}(Q)$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]$$

This means,

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b+c=k}} (1 - p^Q_{abc}) q^{Q,k}_{abc}(Q) + \sum_{\substack{a,b,c \\ a+b+c=1+k}} p^Q_{abc} q^{Q,k}_{abc}(Q)$$

No two bits are equal, because otherwise, the third one would be in $R_1$, and that would violate our assumption on $A$. Since the sum of the three bits is $e_1$, the bits are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x) + B)_{j_1}$, $(\mathbf{C}(x) + B)_{j_2}$, and $(\mathbf{C}(x) + B)_{j_3}$ are three independent, uniformly random bits. Thus, $\forall k, a, b, c$: $q^{Q,k}_{abc}(Q) = \frac{1}{8}$. So, in the even $k$ case,

$$Err_{Q,0} = Err_{Q,2} = \Big((1 - p^Q_{000}) + (1 - p^Q_{011}) + (1 - p^Q_{110}) + (1 - p^Q_{101}) + p^Q_{100} + p^Q_{010} + p^Q_{001} + p^Q_{111}\Big)/8$$

For simplicity, call $P_Q \triangleq Err_{Q,0}$. In the odd $k$ case,

$$\begin{aligned} Err_{Q,1} = Err_{Q,3} &= \Big(p^Q_{000} + p^Q_{011} + p^Q_{110} + p^Q_{101} + \\ &\qquad (1 - p^Q_{100}) + (1 - p^Q_{010}) + (1 - p^Q_{001}) + (1 - p^Q_{111})\Big)/8 \\ &= 1 - P_Q \end{aligned}$$

By Lemma 5.2.2, we know the following two things:

$$\begin{aligned}
&\Pr_B[|B \cap Q| = 0] + \Pr_B[|B \cap Q| = 2] \\
&\qquad \geq (1-\beta)^3 - \frac{9}{\gamma m} + 3\beta^2(1-\beta) - \frac{9}{\gamma m} \\
&\qquad \geq (1-\beta)^3 + 3\beta^2(1-\beta) - \frac{18}{\gamma m} \\
&\qquad \geq (1-\beta)^3 + 3\beta^2(1-\beta) - \frac{36}{m} \\
&\Pr_B[|B \cap Q| = 1] + \Pr_B[|B \cap Q| = 3] \\
&\qquad \geq 3\beta(1-\beta)^2 - \frac{9}{\gamma m} + \beta^3 - \frac{9}{\gamma m} \\
&\qquad \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}
\end{aligned}$$

Combining everything, we find

$$\begin{aligned}
Err_Q &\geq \Big((1-\beta)^3 + 3\beta^2(1-\beta) - \frac{36}{m}\Big)P_Q + \\
&\qquad \Big(3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m}\Big)(1-P_Q) \\
&= 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} + \\
&\qquad \Big((1-\beta)^3 + 3\beta^2(1-\beta) - 3\beta(1-\beta)^2 - \beta^3\Big)P_Q \\
&= 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} + \Big(((1-\beta)-\beta)^3\Big)P_Q
\end{aligned}$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} \geq 2\beta(1-\beta) - \frac{36}{m}$.

- $e_1$ is in the span of two of the vectors representing the bits, say $a_{j_1}$ and $a_{j_2}$, taken by themselves; and $a_{j_3}$ is different from $a_{j_1}$ and $a_{j_2}$: We can

decompose $Err_Q$ into

$$\begin{aligned} Err_Q = \sum_{k=0}^{2} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k] \cdot \\ \Pr_{B}[|B \cap \{j_1, j_2\}| = k \mid A \text{ queries } Q] \\ = \sum_{k=0}^{2} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k] \cdot \\ \Pr_{B}[|B \cap \{j_1, j_2\}| = k] \end{aligned}$$

Note that for any $Q$, $j_1, j_2 \in Q$, and $0 \leq k \leq 2$, the events $A$ queries $Q$ and $|B \cap \{j_1, j_2\}| = k$ are independent. So for any $Q$, $j_1, j_2 \in Q$, and $0 \leq k \leq 2$, $\Pr[A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k] > 0$. Thus, above we are conditioning on events with nonzero probability. The second equality above also holds because of the independence of $A$ queries $Q$ and $|B \cap \{j_1, j_2\}| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k]$. We can further decompose $Err_{Q,k}$ into

$$\begin{aligned} Err_{Q,k} = \sum_{a,b,c} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k; \\ (\mathbf{C}(x) + B)_Q = abc] \cdot \\ \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = abc \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k] \end{aligned}$$

The sum of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$ is $x_1$. So $a + b = x_1 + |B \cap \{j_1, j_2\}|$, and

the above becomes:

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b=k}} q_{abc}^{Q,k}(\{j_1, j_2\}) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k; (\mathbf{C}(x) + B)_Q = abc]$$
$$+ \sum_{\substack{a,b,c \\ a+b=1+k}} q_{abc}^{Q,k}(\{j_1, j_2\}) \cdot$$
$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap \{j_1, j_2\}| = k; (\mathbf{C}(x) + B)_Q = abc]$$

The event $|B \cap \{j_1, j_2\}| = k$ does not depend on the internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b=k}} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abc] \cdot$$
$$q_{abc}^{Q,k}(\{j_1, j_2\}) +$$
$$\sum_{\substack{a,b,c \\ a+b=1+k}} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = abc] \cdot$$
$$q_{abc}^{Q,k}(\{j_1, j_2\})$$

This means,

$$Err_{Q,k} = \sum_{\substack{a,b,c \\ a+b=k}} (1 - p_{abc}^{Q}) q_{abc}^{Q,k}(\{j_1, j_2\}) + \sum_{\substack{a,b,c \\ a+b=1+k}} p_{abc}^{Q} q_{abc}^{Q,k}(\{j_1, j_2\})$$

$a_{j_1}$ and $a_{j_2}$ cannot be equal because they sum to $e_1$, and the sum of all three bits cannot be zero because $j_3$ is not in $R_1$. Thus, $a_{j_1}$, $a_{j_2}$, and $a_{j_3}$ are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x) + B)_{j_1}$, $(\mathbf{C}(x) + B)_{j_2}$, and $(\mathbf{C}(x) + B)_{j_3}$ are three

independent, uniformly random bits. Thus, $\forall k, a, b, c$: $q_{abc}^{Q,k}(\{j_1, j_2\}) = \frac{1}{8}$. So, in the even $k$ case,

$$\begin{aligned} Err_{Q,0} &= Err_{Q,2} \\ &= \Big((1 - p_{000}^Q) + (1 - p_{001}^Q) + (1 - p_{110}^Q) + (1 - p_{111}^Q) + \\ &\qquad p_{100}^Q + p_{101}^Q + p_{010}^Q + p_{011}^Q\Big)/8 \end{aligned}$$

For simplicity, call $P_Q \triangleq Err_{Q,0}$. In the $k = 1$ case,

$$\begin{aligned} Err_{Q,1} &= \Big(p_{000}^Q + p_{001}^Q + p_{110}^Q + p_{111}^Q + \\ &\qquad (1 - p_{100}^Q) + (1 - p_{101}^Q) + (1 - p_{010}^Q) + (1 - p_{011}^Q)\Big)/8 \\ &= 1 - P_Q \end{aligned}$$

By Lemma 5.2.2, we know the following two things:

$$\begin{aligned} \Pr_B[|B \cap \{j_1, j_2\}| = 0] + \Pr_B[|B \cap \{j_1, j_2\}| = 2] &\geq (1 - \beta)^2 - \frac{4}{\gamma m} + \beta^2 - \frac{4}{\gamma m} \\ &\geq (1 - \beta)^2 + \beta^2 - \frac{8}{\gamma m} \\ &\geq (1 - \beta)^2 + \beta^2 - \frac{16}{m} \\ \Pr_B[|B \cap \{j_1, j_2\}| = 1] &\geq 2\beta(1 - \beta) - \frac{4}{\gamma m} \\ &\geq 2\beta(1 - \beta) - \frac{8}{m} \end{aligned}$$

Combining everything, we find

$$\begin{aligned} Err_Q &\geq \Big((1-\beta)^2+\beta^2-\frac{16}{m}\Big)P_Q + \Big(2\beta(1-\beta)-\frac{8}{m}\Big)(1-P_Q) \\ &\geq \min\Big((1-\beta)^2+\beta^2-\frac{16}{m}, 2\beta(1-\beta)-\frac{8}{m}\Big) \\ &> \min\Big((1-\beta)^2+\beta^2, 2\beta(1-\beta)\Big)-\frac{16}{m} \end{aligned}$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 2\beta(1-\beta)-\frac{16}{m}$.

- $e_1$ is in the span of two of the vectors representing the bits, say $a_{j_1}$ and $a_{j_2}$, taken by themselves; and $\mathbf{C}(x)_{j_3}$ is the same as one of the other bits – assume it is $\mathbf{C}(x)_{j_1}$: Define $Z$ as the event that either both $j_1$ and $j_3$ are corrupted or neither are. Consider the decomposition:

$$\begin{aligned} &Err_Q \\ &= \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z} \mid A \text{ queries } Q] + \\ &\qquad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z] \Pr_B[Z \mid A \text{ queries } Q] \\ &= \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z} \mid A \text{ queries } Q] + \\ &\qquad \Big(\sum_{k=0}^{2} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k] \cdot \\ &\qquad\qquad \Pr_B[|B \cap \{j_1, j_2\}| = k \mid A \text{ queries } Q; Z]\Big) \cdot \\ &\qquad\qquad \Pr_B[Z \mid A \text{ queries } Q] \end{aligned}$$

Note that for any $Q$, the events $A$ queries $Q$ and $Z$ are independent. So for any $Q$, $\Pr[A \text{ queries } Q; Z] > 0$. Thus, above we are conditioning on events with nonzero probability. For any $Q$ and $k$, $\Pr[|B \cap \{j_1, j_2\}| =$

$k; A$ queries $Q; Z] > 0$, so we are not conditioning on zero probability events after the second equality, either. Because the events $Z$ and $A$ queries $Q$ are independent, the expression above becomes

$$Err_Q = \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z}] + \Big(\sum_{k=0}^{2} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k] \cdot \Pr_B[|B \cap \{j_1, j_2\}| = k \mid A \text{ queries } Q; Z]\Big) \Pr_B[Z]$$

Let us first consider the error conditioned on $\bar{Z}$. For simplicity, define, $Err_{Q,\bar{Z}} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}]$. We can further decompose $Err_{Q,\bar{Z}}$ into

$$Err_{Q,\bar{Z}} = \sum_{a,b,c \neq a} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; (\mathbf{C}(x)+B)_Q = abc] \cdot \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; \bar{Z}]$$

For simplicity, let us define:

$$q_{abc}^{Q,\bar{Z}} \triangleq \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; \bar{Z}]$$

So the above becomes:

$$
\begin{aligned}
&Err_{Q,\bar{Z}} \\
&= \sum_{a,b,c\neq a} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; (\mathbf{C}(x)+B)_Q = abc] q_{abc}^{Q,\bar{Z}} \\
&= \sum_{a,b,c\neq a} \Big( \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] \cdot \\
&\qquad \Pr_{x,B}[j_1 \in B \mid A \text{ queries } Q; \bar{Z}; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc] \cdot \\
&\qquad \Pr_{x,B}[j_3 \in B \mid A \text{ queries } Q; \bar{Z}; (\mathbf{C}(x)+B)_Q = abc] \Big) q_{abc}^{Q,\bar{Z}}
\end{aligned}
$$

If $A$ queries $Q$, $\bar{Z}$, and $(\mathbf{C}(x)+B)_Q = abc$, over random $x$ and $B$, then it is equally likely that $j_1 \in B$ or $j_3 \in B$. So

$$
\begin{aligned}
Err_{Q,\bar{Z}} = \sum_{a,b,c\neq a} &\frac{1}{2} q_{abc}^{Q,\bar{Z}} \Big( \\
&\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc] \Big)
\end{aligned}
$$

Now, conditioning on whether $j_2$ is corrupted or not, and using that $a+b$

$= x_1$ when $j_1$ and $j_2$ are uncorrupted:

$$
\begin{aligned}
&Err_{Q,\bar{Z}} = \\
&\sum_{a,b,c\neq a} \frac{1}{2}\Big(\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q;\bar{Z};j_1,j_2 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q;\bar{Z};j_1 \in B;(\mathbf{C}(x)+B)_Q = abc]+ \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q;\bar{Z};j_1 \in B;j_2 \notin B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q;\bar{Z};j_1 \in B;(\mathbf{C}(x)+B)_Q = abc]+ \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q;\bar{Z};j_2,j_3 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q;\bar{Z};j_3 \in B;(\mathbf{C}(x)+B)_Q = abc]+ \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q;\bar{Z};j_2 \notin B;j_3 \in B; \\
&\qquad\qquad (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q;\bar{Z};j_3 \in B;(\mathbf{C}(x)+B)_Q = abc]\Big) q_{abc}^{Q,\bar{Z}}
\end{aligned}
$$

The events $\bar{Z}$, $j_1 \in B$, $j_2 \in B$, and $j_3 \in B$ are each independent of the

internal randomness of $A$. Therefore, by Lemma 5.2.4,

$$
\begin{aligned}
&Err_{Q,\bar{Z}} \\
&= \sum_{a,b,c\neq a} \frac{1}{2}\Big( \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \cdot \\
&\qquad \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \cdot \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \cdot \\
&\qquad \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \cdot \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc] \Big) q_{abc}^{Q,\bar{Z}}
\end{aligned}
$$

We notice that

$$
\begin{aligned}
&\Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] \\
&\quad = \Pr_{x,B}[j_2 \in B \mid \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] \\
&\quad = \Pr_{x,B}[x_1 = a+b \mid \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] \\
&\quad = \Pr_{x,B}[x_1 = a+b \mid \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = cba] \\
&\quad = \Pr_{x,B}[x_1 = c+b \mid \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc] \\
&\quad = \Pr_{x,B}[j_2 \in B \mid \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc] \\
&\quad = \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc]
\end{aligned}
$$

Likewise, it is also true that

$$\Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] =$$
$$\Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_3 \in B; (\mathbf{C}(x)+B)_Q = abc]$$

Thus, we obtain

$$\begin{aligned}
&Err_{Q,\bar{Z}} \\
&= \sum_{a,b,c\neq a} \frac{1}{2}\Big(\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\quad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]\cdot \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc]\Big) q_{abc}^{Q,\bar{Z}} \\
&= \sum_{a,b,c\neq a} \frac{1}{2}\Big(\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] + \\
&\qquad \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) = a+b \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc]\Big)\cdot \\
&\quad \Big(\Pr_{x,B}[j_2 \in B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc] + \\
&\qquad \Pr_{x,B}[j_2 \notin B \mid A \text{ queries } Q; \bar{Z}; j_1 \in B; (\mathbf{C}(x)+B)_Q = abc]\Big) q_{abc}^{Q,\bar{Z}} \\
&= \sum_{a,b,c\neq a} \frac{1}{2} q_{abc}^{Q,\bar{Z}} \\
&= \frac{1}{2}
\end{aligned}$$

Also, Lemma 5.2.2 says $\Pr_B[\bar{Z}]$ is at least $2\beta(1-\beta)-\frac{4}{\gamma m} \geq 2\beta(1-\beta)-\frac{8}{m}$.

Let us now consider the error conditioned on $Z$. For simplicity, define, $Err_{Q,Z,k} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k]$. We can further decompose $Err_{Q,Z,k}$ into

$$Err_{Q,Z,k} = \sum_{a,b,c=a} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; \\ |B \cap \{j_1, j_2\}| = k; (\mathbf{C}(x)+B)_Q = abc] \cdot \\ \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k]$$

For simplicity, let us define:

$$q_{abc}^{Q,Z,k} \triangleq \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = abc \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k]$$

Note that $a + b = x_1 + (k \bmod 2)$. So the above becomes:

$$Err_{Q,Z,k} = \\ \sum_{\substack{a,b,c=a \\ a+b=k \bmod 2}} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k; \\ (\mathbf{C}(x)+B)_Q = abc] q_{abc}^{Q,Z,k} + \\ \sum_{\substack{a,b,c=a \\ a+b=1+k \bmod 2}} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k; \\ (\mathbf{C}(x)+B)_Q = abc] q_{abc}^{Q,Z,k}$$

The event $Z \cap |B \cap \{j_1, j_2\}| = k$ does not depend on the internal ran-

domness of $A$. Therefore, by Lemma 5.2.4,

$$\begin{aligned} Err_{Q,Z,k} = & \sum_{\substack{a,b,c=a \\ a+b=k \bmod 2}} q_{abc}^{Q,Z,k} \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \\ & + \sum_{\substack{a,b,c=a \\ a+b=1+k \bmod 2}} q_{abc}^{Q,Z,k} \cdot \\ & \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = abc] \end{aligned}$$

This means,

$$Err_{Q,Z,k} = \sum_{\substack{a,b,c=a \\ a+b=k \bmod 2}} (1 - p_{abc}^Q) q_{abc}^{Q,Z,k} + \sum_{\substack{a,b,c \\ a+b=1+k \bmod 2}} p_{abc}^Q q_{abc}^{Q,Z,k}$$

We said earlier that $e_1$ is in the span of $a_{j_1}$ and $a_{j_2}$. Neither $a_{j_1}$ nor $a_{j_2}$ equal $e_1$, so $a_{j_1}$ and $a_{j_2}$ are linearly independent. Since, also, $x$ is uniformly random, Fact 5.1.2 shows that $(\mathbf{C}(x)+B)_{j_1}$ and $(\mathbf{C}(x)+B)_{j_2}$ are two independent, uniformly random bits. Thus, $\forall k, a, b, c = a$: $q_{abc}^{Q,Z,k}$ $= \frac{1}{4}$. So, when $k$ is even,

$$Err_{Q,Z,k} = \Big((1 - p_{000}^Q) + (1 - p_{111}^Q) + p_{101}^Q + p_{010}^Q\Big)/4$$

For simplicity, call this last expression $P_Q$. On the other hand, when $k$ is odd,

$$Err_{Q,Z,k} = \Big(p_{000}^Q + p_{111}^Q + (1 - p_{101}^Q) + (1 - p_{010}^Q)\Big)/4$$

This expression is $1 - P_Q$. We will use these facts after briefly considering another probability expression.

Because the event $A$ queries $Q$ is independent of the operation of the adversary,

$$\Pr_B[|B \cap \{j_1, j_2\}| = k \mid A \text{ queries } Q; Z] = \Pr_B[|B \cap \{j_1, j_2\}| = k \mid Z]$$

By Lemma 5.2.2, we know the following two things:

$$\begin{aligned}
&\Pr_B[|B \cap \{j_1, j_2\}| = 0 \mid Z] \Pr_B[Z] + \Pr_B[|B \cap \{j_1, j_2\}| = 2 \mid Z] \Pr_B[Z] \\
&\qquad \geq (1-\beta)^3 - \frac{9}{\gamma m} + \beta^3 - \frac{9}{\gamma m} \\
&\qquad \geq (1-\beta)^3 + \beta^3 - \frac{36}{m} \\
&\Pr_B[|B \cap \{j_1, j_2\}| = 1 \mid Z] \Pr_B[Z] \\
&\qquad \geq \beta(1-\beta)^2 + \beta^2(1-\beta) - \frac{9}{\gamma m} \\
&\qquad \geq \beta(1-\beta) - \frac{18}{m}
\end{aligned}$$

Combining everything, we find

$$\begin{aligned}
Err_Q = &\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; \bar{Z}] \Pr_B[\bar{Z}] + \\
&\sum_{k=0}^{2} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; Z; |B \cap \{j_1, j_2\}| = k] \cdot \\
&\qquad \Pr_B[|B \cap \{j_1, j_2\}| = k \mid Z] \Pr_B[Z] \\
\geq &\Big(2\beta(1-\beta) - \frac{8}{m}\Big)\frac{1}{2} + \Big((1-\beta)^3 + \beta^3 - \frac{36}{m}\Big) P_Q + \\
&\qquad \Big(\beta(1-\beta) - \frac{18}{m}\Big)(1 - P_Q) \\
\geq &\Big(2\beta(1-\beta) - \frac{8}{m}\Big)\frac{1}{2} + \min\Big((1-\beta)^3 + \beta^3 - \frac{36}{m}, \beta(1-\beta) - \frac{18}{m}\Big) \\
\geq &\ \beta(1-\beta) + \min\Big((1-\beta)^3 + \beta^3, \beta(1-\beta)\Big) - \frac{40}{m}
\end{aligned}$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq 2\beta(1-\beta) - \frac{40}{m}$.

Since for all $Q$, $Err_Q \geq 2\beta(1-\beta) - \frac{40}{m}$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 2\beta(1-\beta) - \frac{40}{m}$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 2\beta(1-\beta) - \frac{40}{m}$ (where the probability is only over the internal coin flips of $A$).

Remember that $\beta = \frac{\delta - \frac{|R_1|}{m}}{\gamma}$. First note that the expression $2\beta(1-\beta)$ is strictly increasing for $\beta \leq \frac{1}{2}$. Therefore, we can lower bound $2\beta(1-\beta) - \frac{40}{m}$ evaluated at $\beta = \frac{\delta - \frac{|R_1|}{m}}{\gamma}$ with $2\hat{\beta}(1-\hat{\beta}) - \frac{40}{m}$ evaluated at (note: $|R_1| \leq \frac{m}{n}$) $\hat{\beta} = \delta - \frac{1}{n}$:

$$\begin{aligned} 2(\delta - \frac{1}{n})(1 - (\delta - \frac{1}{n})) - \frac{40}{m} &\geq 2\delta(1-\delta) - \frac{6}{n} - \frac{40}{m} \\ &\geq 2\delta(1-\delta) - \frac{46}{n} \end{aligned}$$

The last line holds because the lower bound of Katz and Trevisan [32] implies, for large enough $n$, $m \geq n$. ■

## 10.3 Two Query, Binary, Possibly Non-Linear LDCs

**Claim 10.3** *Let* $\mathbf{C}$*:* $\{0,1\}^n \rightarrow \{0,1\}^m$ *be a code with large enough* $n$*. Fix* $t$ *and let* $\nu \triangleq \frac{1}{.99n(1-H(\frac{1}{2}+\frac{t}{2}))}$*. For any two query LDC recovery algorithm* $A$*,* $\zeta(A) \leq 1 - 2\delta(1-\delta) + 2\nu + t + \frac{8}{n}$.

**Proof:** For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \;\middle|\; \left| \Pr_{x \in \{0,1\}^n}[x_i = \mathbf{C}(x)_j] - \Pr_{x \in \{0,1\}^n}[x_i \neq \mathbf{C}(x)_j] \right| > t \right\}$$

Now consider:

$$S \triangleq \left\{ i \in [n] \;\middle|\; |R_i| \geq \nu m \right\}$$

Clearly $|S|\nu m \leq \sum_{i \in [n]} |R_i|$. So there exists a $j \in [m]$ belonging to at least $\nu|S|$ of the $R_i$ sets. Theorem 2 from [32] then proves that $\nu|S| \leq \frac{1}{1-H(\frac{1}{2}+\frac{t}{2})}$. Therefore, $|S| \leq \frac{1}{\nu}\frac{1}{1-H(\frac{1}{2}+\frac{t}{2})} = .99n < n$. So $\bar{S}$ contains at least one $i$. Without loss of generality, $1 \in \bar{S}$. That is, $|R_1| < \nu m$. Consider what happens when the recovery algorithm is tasked to find $x_1$.

Because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries two positions. If the algorithm ever queried fewer than two positions, have it query more and ignore the additional values obtained.

Define $\gamma \triangleq \frac{|[m] \setminus R_1|}{m}$ and $\beta \triangleq \min(\frac{\delta-\nu}{\gamma}, \frac{1}{2})$. Let $A$ be a $q$ query algorithm for $\mathbf{C}$ subjected to $\delta$ fraction of the codeword corrupted. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus R_1$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of $A$. (For emphasis, the adversary

chooses $B_1$ to always have the same size; but, for $B_2$, it chooses whether to include each member of $R_1$ independently.) For convenience, define $B \triangleq B_1 \cup B_2$. Let $\mathbf{C}(x)+B$ denote the codeword for $x$, corrupted in the positions determined by $B$. Note that $|B| \leq \delta m$ always. Because of Lemma 5.2.1, we can, without loss of generality, assume that $A$ never queries $R_1$.

Without loss of generality, we assume that $A$ flips all of its random coins first, and then, based on those random values, chooses a query set $Q \subset [m]$ and a deterministic function $\phi$ to apply on the two values it receives from querying $Q$. Without loss of generality, $Q = \{1, 2\}$. We use the shorthand "$Q$, $\phi$" to mean the event $A$ has chosen to query $Q$ and use function $\phi$. Now consider the decomposition:

$$\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] = \sum_{Q \subset [m]:|Q|=2,\phi} \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid Q, \phi] \Pr[Q, \phi]$$

Define $Err_{Q,\phi} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid Q, \phi]$. We will bound $Err_{Q,\phi}$ using the following concept. Define the correlation between two Boolean functions $f$ and $g$ as

$$Corr(f, g) \triangleq \Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)]$$

Let us consider the quantity $|Corr(x_i, \phi(Y_1, Y_2))|$. With $\chi_S(Y_1, Y_2) \triangleq \sum_{s \in S} Y_s$ for $S \subseteq \{1, 2\}$, Lemma 6.2.3 for three query LDCs also readily shows

$$\begin{aligned}
&\Big|Corr(x_i, \phi(Y_1, Y_2))\Big| \\
&\leq \Big|Corr(x_i, 0)\Big| + \sum_{S \subseteq \{1,2\} \,:\, |S|=1} \Big|Corr(x_i, \chi_S(Y_1, Y_2))\Big| + \Big|Corr(x_i, Y_1 + Y_2)\Big|
\end{aligned}$$

The first term of this expression is 0 because $\Pr_x[x_i = 0] = \frac{1}{2}$. The two absolute values in the second term are each at most $t$. This is because for any $j \in [m]$, if $|Corr(x_i, \mathbf{C}(x)_j)| > t$, then $j_1$ is corrupted by $B$ into a uniformly random value in $\{0, 1\}$. Therefore, the correlation of the corrupted value with $x_i$ is 0. This gives:

$$\Big|Corr(x_i, \phi(Y_1, Y_2))\Big| \leq 2t + \Big|Corr(x_i, Y_1 + Y_2)\Big|$$

For simplicity, let us temporarily just operate on $|Corr(x_i, Y_1 + Y_2)|$:

$$\begin{aligned}
\Big|Corr(x_i, Y_1 + Y_2)\Big| &= \Big|\Pr[x_i = Y_1 + Y_2] - \Pr[x_i \neq Y_1 + Y_2]\Big| \\
&= \Big|\Pr[0 = x_i + Y_1 + Y_2] - \Pr[0 \neq x_i + Y_1 + Y_2]\Big|
\end{aligned}$$

Because of the independence of $x$ and $B$, we have:

$$
\begin{aligned}
&= \Big| \Pr[0 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2] \Pr[0 = B_1 + B_2] + \\
&\qquad \Pr[1 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2] \Pr[1 = B_1 + B_2] - \\
&\qquad \Pr[0 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2] \Pr[1 = B_1 + B_2] - \\
&\qquad \Pr[1 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2] \Pr[0 = B_1 + B_2] \Big| \\
&= \Big| (\Pr[0 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2] - \Pr[1 = x_i + \mathbf{C}(x)_1 + \mathbf{C}(x)_2]) \\
&\qquad (\Pr[0 = B_1 + B_2] - \Pr[1 = B_1 + B_2]) \Big| \\
&= \Big| Corr(x_i, \mathbf{C}(x)_1 + \mathbf{C}(x)_2) Corr(0, B_1 + B_2) \Big| \\
&\leq \Big| Corr(x_i, \mathbf{C}(x)_1 + \mathbf{C}(x)_2) \Big| \Big| Corr(0, B_1 + B_2) \Big| \\
&\leq \Big| Corr(0, B_1 + B_2) \Big|
\end{aligned}
$$

Each member of $B$ has been corrupted with probability at least $\beta$. By Lemma 5.2.2, we know:

$$\Pr_B[|B \cap Q| = 1] \geq 2\beta(1-\beta) - \frac{4}{\gamma m} \geq 2\beta(1-\beta) - \frac{8}{m}$$

Because $\beta \leq \frac{1}{2}$, we have

$$
\begin{aligned}
\Big| Corr(x_i, \phi(Y_1, Y_2)) \Big| &\leq 2t + \Big( (1 - \Pr_B[|B \cap Q| = 1] \Big) - \Big( \Pr_B[|B \cap Q| = 1] ) \Big) \\
&= 2t + 1 - 2\Big( \Pr_B[|B \cap Q| = 1] \Big) \\
&\leq 2t + 1 - 2\Big( 2\beta(1-\beta) - \frac{8}{m} \Big)
\end{aligned}
$$

Noting that $Err_{Q,\phi} \leq \frac{1}{2}$ or else the algorithm would just guess randomly, we have:

$$(1 - Err_{Q,\phi}) - Err_{Q,\phi} = \Big|(1 - Err_{Q,\phi}) - Err_{Q,\phi}\Big| = \Big|Corr(x_i, \phi(Y_1, Y_2))\Big|$$
$$\Rightarrow Err_{Q,\phi} \geq 2\beta(1-\beta) - t - \frac{8}{m}$$

Since for all $Q$ and $\phi$, $Err_{Q,\phi} \geq 2\beta(1-\beta) - t - \frac{8}{m}$, $\Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 2\beta(1-\beta) - t - \frac{8}{m}$. Thus, there exists an $x$ and $B$ such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq 2\beta(1-\beta) - t - \frac{8}{m}$, where the probability is only over the internal coin flips of $A$.

Remember that $\beta = \min(\frac{\delta-\nu}{\gamma}, \frac{1}{2})$. When $\beta = \frac{\delta-\nu}{\gamma}$, first note that the expression $2\beta(1-\beta)$ is strictly increasing in $\beta$. Therefore, we can lower bound $2\beta(1-\beta) - t - \frac{8}{m}$ evaluated at $\beta = \frac{\delta-\nu}{\gamma}$ with $2\hat{\beta}(1-\hat{\beta}) - t - \frac{8}{m}$ evaluated at $\hat{\beta} = \delta - \nu$:

$$2\Big(\delta - \nu\Big)\Big(1 - \delta + \nu\Big) - t - \frac{8}{m} \geq 2\delta(1-\delta) - 2\nu - t - \frac{8}{m}$$

The lower bound of Katz and Trevisan [32] implies that $m > n$, for large enough $n$. Therefore, this expression is more than $2\delta(1-\delta) - 2\nu - t - \frac{8}{n}$.

When $\beta = \frac{1}{2}$, $2\beta(1-\beta) - t - \frac{8}{m} = \frac{1}{2} - t - \frac{8}{n}$ for large enough $n$ (again, note that $m > n$). ■

# Chapter 11

# Locally Decodable Erasure Codes

## 11.1 Definitions and Properties

We will consider a variant of Locally Decodable Codes called Locally Decodable Erasure Codes (LDECs). LDECs were first defined in Lu et al. [36] and are similar to normal LDCs. Before defining LDECs here, we will make a third definition that is conceptually halfway in between the definitions of LDCs and LDECs. This will facilitate understanding of the similarities and differences among all the models. This definition is based on the classic papers Katz and Trevisan [32] and Goldreich et al. [24].

**Definition 11.1** *(See [32] and [24].) Define $*$ as an extra alphabet symbol (to represent what are called erasures). For a natural number $q$ and positive reals $\delta$ and $\zeta$, we say that $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ is a $(q, \delta, \zeta)$-PROBABILISTIC LOCALLY DECODABLE ERASURE CODE (Probabilistic LDEC) if there exists a probabilistic oracle machine $A$ such that:*

- *In every invocation, $A$ makes at most $q$ queries. Query $j \in [m]$ to the oracle $y \in (\Gamma \cup \{*\})^m$ is answered by $y_j$. (Think of $y$ as the partially erased codeword that $A$ is examining.)*

- *For every* $i \in [n]$ *and* $x \in \Sigma^n$*, we have* $\Pr[A^{\mathbf{C}(x)}(i) = x_i] = 1$.
- *For every* $i \in [n]$*,* $x \in \Sigma^n$*, and* $y \in (\Gamma \cup \{*\})^m$*, with* $y$ *differing from* $\mathbf{C}(x)$ *in at most* $\delta m$ *locations all of which are* $*$*, the probability* $A^y(i)$ *does not query any* $*$ *positions is at least* $\zeta$.

*The probabilities are taken over the internal coin tosses of* $A$*.* $A$ *is called the DECODING ALGORITHM.*

Note that the functionality of $A$ on a corrupted codeword is not defined, strictly speaking.

Probabilistic LDECs are meant to be the analog of LDCs under the erasure error model. Now here is a rephrasing of [36]'s original definition for LDECs.

**Definition 11.2** *([36].) For a natural number* $q$ *and positive real* $\delta$*, we say that* $\mathbf{C}$*:* $\Sigma^n \rightarrow \Gamma^m$ *is a* $(q, \delta)$*-LOCALLY DECODABLE ERASURE CODE (LDEC) if, for every* $S \subset [m]$ *with* $|S| \leq \delta m$*, there exists* $\{j_1, j_2, ...j_q\} \subset [m] \backslash S$ *and function* $f$ *such that, for every* $x \in \Sigma^n$*,* $f(\mathbf{C}(x)_{j_1}, \mathbf{C}(x)_{j_2}, ...\mathbf{C}(x)_{j_q}) = x_i$.

Informally, an LDEC must support recovery of every input position given only any $1 - \delta$ fraction of the codeword positions. Here is an equivalent definition, based on the classic papers [32] and [24], showing how LDECs are in some sense non-deterministic:

**Definition 11.3** *([36]. See also [32] and [24].) Define $*$ as an extra alphabet symbol (to represent what are called erasures). For a natural number $q$ and positive real $\delta$, we say that $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ is a $(q, \delta)$-LOCALLY DECODABLE ERASURE CODE (LDEC) if there exists a non-deterministic oracle machine $A$ such that:*

- *In every invocation, in each execution path, $A$ makes at most $q$ queries. Query $j \in [m]$ to the oracle $y \in (\Gamma \cup \{*\})^m$ is answered by $y_j$. (Think of $y$ as the partially erased codeword that is examining.)*
- *For every $i \in [n]$ and $x \in \Sigma^n$, in each execution path, we have $A^{\mathbf{C}(x)}(i) = x_i$.*
- *For every $i \in [n]$, $x \in \Sigma^n$, and $y \in (\Gamma \cup \{*\})^m$, with $y$ differing from $\mathbf{C}(x)$ in at most $\delta m$ locations all of which are $*$, there exists at least one execution path of $A^y(i)$ that does not query any $*$ locations.*

*$A$ is called the DECODING ALGORITHM.*

This definition illuminates that the essential difference between Probabilistic LDECs and LDECs is that in the former case, $A$ queries randomly without prior knowledge of where the erasures are, but in the latter case, $A$ somehow knows where the erasures are and can avoid querying them.

Kerenidis and de Wolf [34] gives reductions for LDECs to and from smooth codes, and [32] gives reductions for smooth codes to and from LDCs.

Here are the reductions between LDECs and smooth codes:

**Lemma 11.1** *([34].) A binary $(q, c, \epsilon)$-smooth code is also a binary $(q, \frac{\epsilon}{c} - \frac{1}{m})$-LDEC.*

**Lemma 11.2** *([34].) A binary $(q, \delta)$-LDEC is also a binary $(q, \frac{q}{\delta}, \frac{1}{2})$-smooth code.*

Here we give two lemmas comparing Probabilistic LDECs with LDCs.

**Lemma 11.3** *Let $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ be a $(q, \delta, \zeta)$-Probabilistic LDEC. Then $\mathbf{C}$ is also an $(q, \delta, \epsilon)$-LDC with $\frac{1}{|\Sigma|} + \epsilon \geq \zeta$.*

**Proof:** Because $\mathbf{C}$ is a Probabilistic LDEC, there exists an algorithm $A$ such that for every $i \in [n]$ and $S \subset [m]$ with $|S| \leq \delta m$, the probability $A$ queries an edge intersecting $S$ is less than or equal $1 - \zeta$. Let us now consider $\mathbf{C}$ as an LDC, using the same decoding algorithm $A$. Because for every $i \in [n]$ and $x \in \Sigma^n$, we have $\Pr[A^{\mathbf{C}(x)}(i) = x_i] = 1$ (bullet point two of the definition of a Probabilistic LDEC), $A$ can only be wrong when it queries an edge containing a vertex that has been corrupted. But this happens with no more than $1 - \zeta$ probability. ■

**Lemma 11.4** *Let $\mathbf{C}$ be a $(q, \delta, \epsilon)$-LDC over field $F$ whose recovery algorithm is a linear decoder using exactly $q$ positions. Define $p \triangleq \text{char}(F)$. Then $\mathbf{C}$ is also an $(q, \delta, \zeta)$-Probabilistic LDEC with $\zeta \geq \frac{\epsilon p}{p-1}$.*

**Note:** Requiring a recovery algorithm to be a linear decoder that uses exactly $q$ positions is slightly more restrictive than bullet point 2 in the definition of Probabilistic LDECs.

**Proof:** Let $A$ be a linear decoder using exactly $q$ positions and achieving parameters $(q, \delta, \epsilon)$ on $\mathbf{C}$. We will show that, using $A$ for $\mathbf{C}$ in the Probabilistic LDEC model, we get correctness $\zeta \geq \frac{\epsilon p}{p-1}$. We will prove the contrapositive.

Let $\zeta$ be the correctness $A$ achieves in the Probabilistic LDEC model. Assume $\zeta < \frac{\epsilon p}{p-1}$. This means there exists an $i \in [n]$ and $S \subset [m]$ with $|S| \leq \delta m$ such that if the adversary erases the codeword positions indexed by members of $S$, the probability $A$'s query set does not intersect $S$ is $\zeta$. Arbitrarily fix $x \in F^n$ as the input to the code.

Now let us consider what happens in the LDC model. Have the adversary corrupt the codeword by adding a uniformly random value of $F$ independently to each position indexed by a member of $S$. There are two cases in which $A$ returns the correct answer on this corrupted codeword: either its randomly chosen query set does not intersect $S$ or its randomly chosen query set intersects $S$, but the appropriate linear combination of the random values in the intersection (this is fixed as part of $A$) is 0. The probability $A$ gets the correct answer conditioned on this second case (the randomly chosen query set intersects $S$) occurring is $\frac{1}{p}$. So, the probability $A$ is correct on this corrupted

codeword is

$$\zeta + \frac{1-\zeta}{p} = \zeta(1-\frac{1}{p}) + \frac{1}{p} < \frac{\epsilon p}{p-1}(1-\frac{1}{p}) + \frac{1}{p} = \epsilon + \frac{1}{p}$$

■

## 11.2 Correctness Bounds for Probabilistic LDECs

Just as with our results on LDCs, sometimes with the Probabilistic LDEC model, we will want to emphasize that a proof about the adversary's capabilities holds for all input positions and not just one. To express that, we define a technical variant of correctness:

**Definition 11.4** *Let* $\mathbf{C}$*:* $\Sigma^n \to \Gamma^m$ *be a* $(q, \delta, \zeta)$*-Probabilistic LDEC. Define*

$$\zeta^\star \triangleq \max_A \max_{i \in [n]} \min_{x \in \Sigma^n} \Big( \min_{y \in (\Gamma \cup \{*\})^m \,:\, d(y, \mathbf{C}(x)) \leq \delta m} \Pr[A^y(i) \text{ does not query any } * \text{ positions}] \Big)$$

*where the probability is over the A's internal randomness and the notation* $\max_A$ *means the maximum over all probabilistic oracle machines A such that:*

- *In every invocation, A makes at most q queries. Query* $j \in [m]$ *to the oracle* $y \in (\Gamma \cup \{*\})^m$ *is answered by* $y_j$*. (Think of y as the partially erased codeword that A is examining.)*
- *For every* $i \in [n]$ *and* $x \in \Sigma^n$*, we have* $\Pr[A^{\mathbf{C}(x)}(i) = x_i] = 1$*.*

It is obvious from the definitions that, for every $(q, \delta, \zeta)$-Probabilistic LDEC, $\zeta \leq \zeta^\star$. Here is a fact about the limitations of the performance capabilities of Probabilistic LDECs.

**Claim 11.5** *For any Probabilistic LDEC recovery algorithm $A$ using exactly $q$ positions, $\zeta^\star(A) < (1-\delta)^q$.*

**Proof:** The adversary chooses a set of $\delta m$ positions uniformly at random from $[m]$. The adversary erases these positions. For any given query set the algorithm chooses to query, arbitrarily number the positions 1 to $q$. The probability that the first position is not erased is $\frac{m-\delta m}{m} = 1-\delta$. The probability that the $k$'th position queried ($2 \leq k \leq q$) is not erased conditioned that the first $k-1$ positions were not erased is $\frac{m-k+1-\delta m}{m-k+1} < 1-\delta$. Multiplying these, we find the probability the algorithm is correct with less than $(1-\delta)^q$ probability. ■

# Chapter 12

# Hadamard Codes Can Have Very Small Error

The prototypical example of an LDC is the Hadamard code. In particular, it is possible that, of all LDCs, the Hadamard code achieves the highest $\epsilon$ given any $\delta$ (for a fixed $q$). In fact, we will prove that the Hadamard code gives close to optimal correctness. In this chapter, we investigate the limits of the Hadamard code's performance. Here is the definition of the Hadamard code:

**Definition 12.1** *Fix $x_1, x_2, ...x_n \in \{0,1\}$ as an input. For each $a \in \{0,1\}^n$, define the a'th position of the Hadamard code as $\mathbf{C}_a(x_1...x_n) = a \cdot [x_1...x_n]$.*

As a warmup, we consider the Hadamard code under two query recovery algorithms. This lemma is well known, and it originates from the techniques of Blum et al. [10].

**Lemma 12.1** *When the Hadamard code is used as a two query LDC, the correctness is at least $1 - 2\delta$.*

**Proof:** Without loss of generality, let $x_1$ be the input position requested.

Define $E$ as the following.

$$\Big\{\{a, a+e_1\} \mid a \in \{0,1\}^n\Big\}$$

Define $A$ as the algorithm that selects a member from $E$ to query uniformly at random and sums the two bits it receives. Notice that $E$ is a complete matching over the ground set that is the set of positions in the Hadamard code. So no two members of $E$ intersect. Therefore, for any adversary, each vertex the adversary corrupts can cause at most one edge in $E$ to be recovered incorrectly. Clearly $|E| = \frac{m}{2}$. So, for any $i$, the algorithm errs with probability at most $\frac{\delta m}{\frac{m}{2}} = 2\delta$. ■

Claim 10.1 is very nearly a matching upper bound to this lemma.

Given larger $q$, we can achieve dramatically smaller error:

**Theorem 12.2** *For $\delta < \frac{1}{4}$ and even $q$, there exists a $q$ query recovery algorithm achieving*

$$\zeta_\delta \geq 1 - \Big(4(2\delta)(1-2\delta)\Big)^{\lfloor \frac{q}{4} \rfloor}$$

*for the Hadamard code.*

**Note:** We will actually prove the stronger, but slightly more complicated, bound:

$$\zeta_\delta \geq 1 - 2^{q/2-1}(2\delta)^{\lceil \frac{q}{4} \rceil}(1-2\delta)^{\lfloor \frac{q}{4} \rfloor}$$

**Proof:** Let $z$ be defined so that $q = 2z$. For clarity, call the algorithm used in Claim 12.1 as $\hat{A}$. The algorithm will perform $\hat{A}$ as a sub-procedure $z$ times. Each sub-procedure will use its own random coin flips, and hence each answer produced by an $\hat{A}$ will be independent from the other answers, conditioned on the input to the code and the error. The algorithm will take the majority vote of the $z$ answers it receives. In the case that $z$ is even and the vote is tied, assume the algorithm guesses 0 or 1 with equal probability ($\frac{1}{2}$). For any fixed input to the code and error, let the probability, over the randomness of $\hat{A}$, that the adversary makes one sub-procedure wrong be $\alpha$. The majority vote operation produces the wrong answer when half or more of the sub-procedures return the wrong answer. So the probability of error is:

$$\begin{cases} \sum_{i=0}^{\frac{z-1}{2}} \binom{z}{i}(1-\alpha)^i\alpha^{z-i} & z \text{ odd} \\ \sum_{i=0}^{\frac{z}{2}-1} \binom{z}{i}(1-\alpha)^i\alpha^{z-i} + \frac{1}{2}\binom{z}{z/2}(1-\alpha)^{\frac{z}{2}}\alpha^{\frac{z}{2}} & z \text{ even} \end{cases}$$

We will upper bound these quantities. First note that

$$\begin{cases} \sum_{i=0}^{\frac{z-1}{2}} \binom{z}{i} & z \text{ odd} \\ \sum_{i=0}^{\frac{z}{2}-1} \binom{z}{i} + \frac{1}{2}\binom{z}{z/2} & z \text{ even} \end{cases}$$

both equal $2^{z-1}$. Next, note that Lemma 12.1 shows $\alpha \leq 2\delta$ and, by assumption, $2\delta < \frac{1}{2}$. Therefore, $\frac{\alpha}{1-\alpha} < 1$, and $(1-\alpha)^i\alpha^{z-i}$ is increasing in $i$. Therefore, the error probability is upper bounded by

$$2^{z-1}(1-\alpha)^{\lfloor \frac{z}{2} \rfloor}\alpha^{z-\lfloor \frac{z}{2} \rfloor} = 2^{z-1}(1-\alpha)^{\lfloor \frac{z}{2} \rfloor}\alpha^{\lceil \frac{z}{2} \rceil}$$

This expression is increasing in $\alpha$, and $\alpha \leq 2\delta$, so the result follows. ■

Alternatively, in the last proof, we could have upper bounded the error probability using the Chernoff bound:

$$\sum_{i=0}^{\lfloor \frac{z}{2} \rfloor} \binom{z}{i} (1-\alpha)^i \alpha^{z-i} \leq \sum_{i=0}^{\lfloor \frac{z}{2} \rfloor} \binom{z}{i} (1-2\delta)^i (2\delta)^{z-i}$$

this expression increases with $\alpha$ and $\alpha \leq 2\delta$

$$\leq e^{-2z(1-2\delta-\frac{1}{2})^2} \qquad \text{Chernoff bound (noting } \delta < \frac{1}{4})$$

$$= e^{-q(\frac{1}{2}-2\delta)^2}$$

The region where this upper bound is tightest is near where $\delta = \frac{1}{4}$. But we are usually interested in $\delta$ close to 0, instead. So that is why we used the a different upper bounding method in the theorem above.

The algorithm presented in the last theorem uses a majority vote operation, so it is not a linear decoder. Any $q$ query linear decoder would have error at least $q\delta - o(\delta)$, which is much worse than the algorithm in the last theorem. This shows linear decoders do not perform as well as other algorithms when the number of queries gets large.

This construction should be contrasted with our upper bounds on correctness in Chapters 9 and 10. These results show that the Hadamard code can indeed achieve close to optimal correctness.

# Bibliography


[1] Andris Ambainis. Upper bound on communication complexity of private information retrieval. In *ICALP '97: Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, pages 401–407, London, UK, 1997. Springer-Verlag.

[2] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 21–32, New York, NY, USA, 1991. ACM.

[3] R. Beigel, L. Fortnow, and W. Gasarch. A nearly tight lower bound for private information retrieval protocols, 2003.

[4] Amos Beimel and Yuval Ishai. Information-theoretic private information retrieval: A unified construction (extended abstract). In *Proc. of the 28th International Colloquium on Automata, Languages and Programming, volume 2076 of Lecture Notes in Computer Science*, pages 912–926. Springer, 2001.

[5] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *J. Comput. Syst. Sci.*, 71(2):213–247, 2005.

[6] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-Francois Raymond. Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 261–270, Washington, DC, USA, 2002. IEEE Computer Society.

[7] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. Local list decoding with a constant number of queries. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 715–722, Washington, DC, USA, 2010. IEEE Computer Society.

[8] Avraham Ben-Aroya, Oded Regev, and Ronald de Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and ldcs. In *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 477–486, Washington, DC, USA, 2008. IEEE Computer Society.

[9] Eli Ben-Sasson and Michael Viderman. Towards lower bounds on locally testable codes via density arguments. *ECCC TR10-200*, 2010.

[10] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83, New York, NY, USA, 1990. ACM.

[11] Jop Briet and Ronald de Wolf. Locally decodable quantum codes. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, volume 3 of *Leibniz International Proceedings in Informatics*, pages 219–230, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[12] Mahdi Cheraghchi, Anna Gál, and Andrew Mills. Bounds on correctness for locally decodable codes. Unpublished manuscript, 2011.

[13] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[14] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 2006.

[15] Amit Deshpande, Rahul Jain, T. Kavitha, Satyanarayana V. Lokam, and Jaikumar Radhakrishnan. Lower bounds for adaptive locally decodable codes. *Random Struct. Algorithms*, 27(3):358–378, 2005.

[16] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. In *FOCS*, pages 705–714, 2010.

[17] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 592–601, New York, NY, USA, 2005. ACM Press.

[18] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 39–44, New York, NY, USA, 2009. ACM.

[19] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In *CRYPTO*, pages 205–210, 1982.

[20] Anna Gal and Andrew Mills. Three query linear locally decodable codes with higher correctness require exponential length. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, Leibniz International Proceedings in Informatics, Dortmund, Germany, 2011. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[21] Anna Gal and Andrew Mills. Three query linear locally decodable codes with higher correctness require exponential length. *Transactions on Computation Theory*, 2011.

[22] W. Gasarch. A survey on private information retrieval. In *Bulletin of the EATCS*, volume 82, pages 72–107, 2004.

[23] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 151–160, New York, NY, USA, 1998. ACM.

[24] Oded Goldreich, Howard Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private infor-

mation retrieval. *Comput. Complex.*, 15(3):263–296, 2006.

[25] Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. *J. ACM*, 53(4):558–655, 2006.

[26] Eric Hielscher. A survey of locally decodable codes and private information retrieval schemes, 2007.

[27] Yuval Ishai and Eyal Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In *In Proc. of 31st STOC*, pages 79–88, 1999.

[28] Toshiya Itoh. Efficient private information retrieval. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E82-A(1):11–20, January 1999.

[29] Toshiya Itoh. On lower bounds for the communication complexity of private information retrieval. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E84-A(1), January 2001.

[30] Rahul Jain. Towards a classical proof of exponential lower bound for 2-probe smooth codes. 2006.

[31] Yael Tauman Kalai and Ran Raz. Succinct non-interactive zero-knowledge proofs with preprocessing for logsnp. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 355–366, Washington, DC, USA, 2006. IEEE Computer Society.

[32] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC '00: Proceedings of the 32nd annual ACM Symposium on Theory of Computing*, pages 80–86, New York, NY, USA, 2000. ACM Press.

[33] Kiran S. Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of mersenne numbers. In *IEEE Conference on Computational Complexity*, pages 175–186, 2008.

[34] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *STOC '03: Proceedings of the thirty-fifth Annual ACM Symposium on Theory of Computing*, pages 106–115, New York, NY, USA, 2003. ACM Press.

[35] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 167–176, New York, NY, USA, 2011. ACM.

[36] Chi-Jen Lu, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Extractors: optimal up to constant factors. In *STOC*, pages 602–611. ACM, 2003.

[37] E. Mann. Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1998.

[38] Kenji Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *RANDOM '02: Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 39–50, London, UK, 2002. Springer-Verlag.

[39] Prasad Raghavendra. A note on Yekhanin's locally decodable codes. *ECCC TR07-016*, 2007.

[40] Alexander A. Razborov and Sergey Yekhanin. An $\Omega(n^{1/3})$ lower bound for bilinear group based private information retrieval. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 739–748, Washington, DC, USA, 2006. IEEE Computer Society.

[41] Alex Samorodnitsky. An attempt to de-quantify the lower bound for 2-query locally decodable code. 2006.

[42] Dungjade Shiowattana and Satyanarayana V. Lokam. An optimal lower bound for 2-query locally decodable linear codes. *Inf. Process. Lett.*, 97(6):244–250, 2006.

[43] Luca Trevisan. Some applications of coding theory in computational complexity. *ECCC TR04-043*, 2004.

[44] S. Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *Proceedings of the 32nd ICALP*, volume 3580 of *LNCS*, pages 1424–1436, 2005.

[45] David Woodruff. Some new lower bounds for general locally decodable codes. *ECCC TR07-006*, 2006.

[46] David Woodruff. Corruption and recovery-efficient locally decodable codes. In *APPROX '08 / RANDOM '08: Proceedings of the 11th international workshop, APPROX 2008, and 12th international workshop, RANDOM 2008 on Approximation, Randomization and Combinatorial Optimization*, pages 584–595, Berlin, Heidelberg, 2008. Springer-Verlag.

[47] David Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *APPROX '10 / RANDOM '10*, 2010.

[48] David Woodruff and Sergey Yekhanin. A geometric approach to information-theoretic private information retrieval. In *CCC '05: Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 275–284, Washington, DC, USA, 2005. IEEE Computer Society.

[49] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. In *STOC '07: Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 266–274, New York, NY, USA, 2007. ACM.

[50] Sergey Yekhanin. *Locally decodable codes*. NOW Publishers, 2010.

# Vita

Andrew Jesse Mills earned a Bachelor of Science degree in Applied Mathematics, Physics, and Computer Science from the California Institute of Technology in 2003.

Permanent address: 3131 Castleleigh Rd; Silver Spring, MD 20904

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.